NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE EFFECT OF TASK COMPLEXITY ON USER INTERFACES: A COMPARISON OF COMMAND LANGUAGE INTERFACE AND DIRECT MANIPULATION INTERFACE

by

Nancy A. Reinhard

MARCH 1991

Thesis Advisor: Kishore Sengupta

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If applicable)* AS	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No. / Work Unit Accession Number

11. TITLE *(Include Security Classification)*
The Effect of Task Complexity on User Interfaces: A Comparison of Command Language Interface and Direct Manipulation Interface

12. PERSONAL AUTHOR(S) Reinhard, Nancy A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	From To	March 1991	~~134~~ 135

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
User-Interface; Human-Computer Interaction; Task Complexity; Direct Manipulation Interface; Command Language Interface

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

A computer-user interface is the software that communicates the user's inputs to the computer and returns information from the computer back to the user. A variety of user interfaces have been developed, including command language interfaces, direct manipulation interfaces, and menus. This research explored the relative benefits of command language interfaces (CLI) and direct manipulation interfaces (DMI) for performance of simple and complex tasks by novices. Two levels of task complexity were used, one requiring five inputs (simple) and one requiring 24 inputs (complex). Dependent variables were (1) time to complete each task set, (2) number of errors, and (3) number of references to on-line help. Results indicate that learning to use a DMI takes longer than learning to use a CLI. No significant difference was observed in time to complete the simple task. However, once a novice learns to use a DMI, a complex task requires less time, fewer errors are made, and references to help screens are required less often. With complex tasks, direct manipulation interfaces appear to help novices to be more productive than do command language interfaces.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Sengupta, Kishore	22b. TELEPHONE *(Include Area code)* (408) 646-3212 22c. OFFICE SYMBOL Code AS/SE

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

Approved for public release; distribution is unlimited.

The Effect of Task Complexity on User Interfaces: A Comparison of Command Language Interface and Direct Manipulation Interface

by

Nancy A. Reinhard
Lieutenant, United States Navy, Reserve
B.S., Wright State University , 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN (Information Systems)

from the

ABSTRACT

A computer-user interface is the software that communicates the user's inputs to the computer and returns information from the computer back to the user. A variety of user interfaces have been developed, including command language interfaces, direct manipulation interfaces, and menus. This research explored the relative benefits of command language interfaces (CLI) and direct manipulation interfaces (DMI) for performance of simple and complex tasks by novices. Two levels of task complexity were used, one requiring five inputs (simple) and one requiring 24 inputs (complex). Dependent variables were (1) time to complete each task set, (2) number of errors, and (3) number of references to on-line help. Results indicate that learning to use a DMI takes longer than learning to use a CLI. No significant difference was observed in time to complete the simple task. However, once a novice learns to use a DMI, a complex task requires less time, fewer errors are made, and references to help screens are required less often. With complex tasks, direct manipulation interfaces appear to help novices to be more productive than do command language interfaces.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

1. *User Interfaces and Complexity*

User interface is described as the dialogue that occurs between a user and a computer in order to accomplish a specific task. The computer interface includes all of the hardware and software needed to support the dialogue. The human or user interface also includes the individual's knowledge and understanding of the system, perception of the system, and motor capabilities. These three factors strongly influence the user's ability to operate the computer satisfactorily. [Ref 1:pp. 23-24]

Card and others describe the design of a human-computer interface in terms of three types of mental processing systems: perceptual, cognitive, and motor. They suggest that the designer of an interface can change the overall level of difficulty of an interface by adjusting input and output difficulty related to the various types of processing. [Ref 1:p. 24]

Card and his colleagues use what is referred to as the GOMS model to describe and predict human behavior (see Appendix A for definitions). The GOMS model describes the user's cognitive structure as consisting of four components: (1) a set of Goals, (2) a set of Operators, (3) a set of Methods for achieving the goals, and (4) a set of Selection

rules for choosing among competing methods for goals. [Ref 1:pp. 139-144]

As described by Card and others, a goal is a symbolic structure that defines a state of affairs to be achieved and determines a set of possible methods by which it can be accomplished. Operators are elementary perceptual, motor, or cognitive acts whose execution is necessary to change any aspect of the user's mental state or affect the task environment. Methods describe procedures for accomplishing a goal. Selection rules are used to determine which method should be used to satisfy a given goal or subgoal in a specified context. [Ref 1:pp. 144-146]

<u>Complexity</u> is defined by Rasmussen as the number of goals and processes that must be controlled by an operator or user of a system. The characteristics of the means available for this control also determine complexity. [Ref 2:p. 24].

<u>User interface complexity</u> is the complexity of the entire interface system from a user's point of view. Interface complexity relates to the user's knowledge of the given system or similar systems, and to the task itself. If the user perceives the interface and/or the task to be complex, the user will have difficulty completing the task successfully. [Ref 3:p. 365]

Kieras and Polson expanded the GOMS model to develop a new model, the Production model, that includes production rules and can be used to predict ease of learning and using a

system as a function of complexity. A production rule is a condition-action pair of the form:

IF (condition) THEN (action).

When a system requires the use of many productions rules, it is complex and will be difficult to learn. The Kieras and Polson work is known as the Cognitive Complexity approach. [Ref 3:p. 1]

Bovair and others have conducted further research based on the Cognitive Complexity approach to describe principles for constructing Production models. Their work has provided support for the hypothesis that production rule models can be used to make quantitative predictions related both to ease of learning and to ease of use. [Ref 4:p. 1]

2. Task Complexity

Wood defines <u>task complexity</u> in terms of information processing and decision making [Ref 5:p. 40]. According to Wood, task complexity is the measurable complexity of the task that is being performed. If task inputs and outputs are known, the total task complexity can be calculated. The inputs for a task include the acts required to carry out the task and the information cues about the task that the user has available. These inputs are processed to form a product, which is the output. [Ref 6:pp. 66-74]

Wood identifies the components of a task as products, required acts, and information cues. These components are used to define three forms of task complexity, referred to as

component, coordinative, and dynamic. Component complexity results from the combination of the required acts and the information cues. Coordinative complexity is due to the inputs (required acts and information cues) as the function of the products. Dynamic complexity is the result of changes in the states of the world which affect the relationships between task inputs and products. [Ref 6:pp. 64-74]

Wood proposes that variations in task complexity produce changes in knowledge, skill, and effort requirements for successful task performance. Additionally, task performance depends on the combination of component, coordinative, and dynamic complexity. [Ref 6:p. 79]

3. Command Language and Direct Manipulation Interfaces

Command language (CL) interfaces are the traditional, most common computer interfaces. With these systems, the user communicates with the computer by typing commands, each of which has a specific syntax. These interfaces generally require the user to memorize the command syntax and the order of the commands for successful communication with the computer. As a result, CL interfaces usually are considered to have high user interface complexity. [Ref 7:p. 154]

Direct manipulation (DM) interfaces are newer. They are defined by Jacob to be interfaces that present a set of visual representations on a display and provide a repertoire of

manipulations that can be performed on any of the visual representations. [Ref 8:p. 283]

Using these interfaces, the user performs some form of action directly on an object or other representation (often referred to as an icon) displayed on the computer screen. DM systems are characterized by low levels of interface complexity, requiring only minimal syntactic knowledge and semantic knowledge related to computer use. The user can go directly to the task without understanding complex computer concepts. Since icons are graphical representations of objects or actions to be performed, their purposes and uses can be intuitive if they are well designed. Such DM interfaces require only that the user understand what each icon stands for and what operations are permitted in order to perform an operation successfully. [Ref 9:p. 207]

Novice users prefer DM interfaces over CL interfaces because only the operations and the meanings of the icons must be memorized. On the other hand, CL interfaces are preferred by experienced users because such interfaces are more responsive [Ref 7:p. 155]. Even with experts, however, at some point memorizing complex CL commands is beyond the ability of most users. Thus many software application programs (e.g., dBase IV) provide both forms of interface for their users.

Margono and others compared user performance with DM and CL interfaces for file management and operating system tasks.

They determined that the Apple Macintosh Computer DM file management systems are easier and faster to learn for the novice than systems using the CL interface, for the tasks that were tested. Study participants had limited computer background. [Ref 7:p. 158]

Karat has determined that DM operations have a consistent performance advantage over CL interfaces as task difficulty increases. Karat also determined that DM was easier to use for those participating in his study, who were not experienced in file management systems. [Ref 10:p. 491]

Te'eni compared the feedback provided by a DM interface to that from a CL interface, when students of an information systems management course calculated their final course grades based on midterm, final exam, and class project scores. He determined that DM interfaces enhance cognitive control and require less effort than CL. [Ref 11:p. 22]

No apparent dividing line has been established to determine the conditions under which DM interfaces are preferable to CL interfaces. The user's level of expertise is one determining factor. The level of task complexity is another. A third factor is the overall interface complexity that results from the combination of task requirements and the specific user. That is, the user's pre-existing computer and task knowledge result in a personal perception of interface complexity.

B. GOAL AND SCOPE

The goal of this study is to determine if the user will benefit from the use of a DM interface over a CL interface for simple and complex tasks. Research has been conducted to evaluate the performance differences observed between users working with CL interfaces and those using DM interfaces for two levels of task complexity.

The scope of the study is limited to comparing the relative usefulness of the two types of user interfaces as a function of task complexity. Only two levels of task complexity were considered and conditions of use were such that stress was not a factor in the experiment. The tasks involved in the experiment were limited to computer operating systems tasks.

Only novice users were included in the study. Thus the effects of these variables on performance by experts cannot be inferred from the results. Additionally, the study was limited to 25 subjects, all of whom were masters students at a military institution and not necessarily representative of the general population.

II. COMPLEXITY AND USER INTERFACES

A. USER INTERFACE

The user interface, as described by Card and others, is the equipment used by a human and a computer to engage in a communicative dialogue in order to accomplish a task. The process is considered a dialogue because both the user and the computer have access to the stream of symbols flowing back and forth to accomplish the communication; each can interrupt, query, and correct the communication at various points in the process. All mechanisms used in this dialogue constitute the interface: the physical devices, such as the keyboards and displays, and computer programs for controlling the interaction. [Ref 1:p. 4]

Card and others report that human psychology plays a major role in human-computer interaction and should be considered in the design of a user interface. The three psychology constructs to be considered are perceptual, cognitive, and motor [Ref 1:p. 24]. A designer of a user interface can alter the degree of difficulty of learning and using computer applications with a given interface by varying the level of input and output related to the three interacting human subsystems described by these branches of psychology: the perceptual system, the cognitive system, and the motor system.

1. Perceptual System

The perceptual system, as defined by Card and others, carries sensations of the physical world detected by the body's sensory systems into internal representations of the mind by means of integrated sensory systems. For example, a visual stimulus is observed by the eyes and stored for about 0.2 second in the visual image store. This image store holds an exact representation of what was seen by the eyes while it is being symbolically coded. The visual perceptual system includes central vision, peripheral vision, eye movement, and head movement, as these operate as an integrated system to provide a continual representation of the visual scene to the perceiver. [Ref 1:pp. 25-34]

2. Cognitive System

The cognitive system of an individual provides the ability to move symbolic information obtained from the sensory image stores to working memory, where it is combined with information previously stored in long-term memory [Ref 1:p. 24]. In other words, the cognitive system processes information from the perceptual system so that corresponding actions can be output via the motor system [Ref 1:p. 35]. Cognitive psychology is the science of studying human information processing.

The current model of the cognitive system includes two forms of memory: working memory to hold current information,

and long-term memory to store knowledge for future use [Ref 1:p. 35]. According to Card and others, working memory contains intermediate products of thinking and the representations produced by the perceptual system. Working memory is where all mental operations obtain their operands and leave their outputs. It also consists of a subset of the elements in long-term memory that have been activated. [Ref 1:pp. 36-39]

Long-term memory contains the user's mass of available knowledge. It is a network of related symbols called chunks, accessed associatively from the contents of working memory. Long-term memory contains not only facts, but procedures and history as well. [Ref 1:pp. 39-41]

3. Motor System

The motor system carries out the responses of the perceptual and cognitive systems. Using the motor system, an individual's thoughts are translated into action by activating patterns of voluntary muscles. For computer users the two most important effectors are the arm-hand-finger system and the head-eye system. The human motor system must be considered in developing a user interface to ensure that a desired action can be physically accomplished. [Ref 1:p. 34]

B. INTERFACE COMPLEXITY

1. Definition of Complexity

Rasmussen defines complexity as a combination of the number of goals to be accomplished, the number of processes that must be controlled, and the number and characteristics of the means (devices) available for this control [Ref 2:p. 24]. Complexity, therefore, is a function of two components: the task (i.e., the number of goals/processes) and the user interface (i.e., the number of characteristics or devices available for control).

a. *Task Complexity*

Task complexity in relation to user interfaces is not well defined in the literature. It is apparent that simple tasks have a better chance of being completed successfully than complex tasks. Unfortunately, a cut-off point, i.e., the number of instructions or steps differentiating a complex task from a simple task, has not been defined. Only relative measures of complexity between tasks have been compared. For example, the task of turning on a computer (flipping a switch) is considered simple when compared to editing a text file using commands such as insert, delete, copy, etc. The latter is considered a complex task. Task complexity is described in greater detail later.

b. User Interface Complexity

User interface complexity generally is considered as the overall complexity of a system or device from a user's perspective. Kieras and Polson call this <u>cognitive complexity</u> [Ref 3:p. 365]. This form of complexity is dependent on the amount, content, and structure of the knowledge required to operate the device or system. For a new user, complexity is also determined by the difficulty of acquiring the new knowledge necessary for using the device or system. For the scope of this study, we are considering task complexity but not user interface complexity (i.e., user interface complexity is held at one level). [Ref 3:pp. 364-365]

2. Task and Device Representations

Kieras and Polson identified two major components of the knowledge involved in operating a device as the user's task representation and the user's device representation [Ref 3:p 366]. The user's <u>task representation</u> is the user's knowledge of how to carry out a task, which may be described by the GOMS model developed by Card and others [Ref 1:p 140-146].

<u>Device Representation</u> is the user's knowledge of that device and how it is used to carry out tasks. The complexity of a device is determined by the complexity of knowledge required to operate the device. Kieras and Polson theorize that device complexity depends on three factors:

(1) The complexity of the user's task representation, and the learning, memory, and processing capacity demands implied by the task representation.

(2) The number of device-dependent functions which are not part of the user's initial task representation, and the difficulty of learning them.

(3) The ease with which a user can acquire how-it-works knowledge. [Ref 1:p. 367]

Additionally, Kieras and Polson identify four categories of information related to the user's knowledge of the device.

(1) <u>Task-relevant knowledge</u> is information about the goals that the device can be used to satisfy, the operations that can be performed on the device, and the operating procedures for the device. Such knowledge of the device is a counterpart to the user's task representation.

(2) <u>Device layout knowledge</u> is knowledge concerning the physical layout of the device, such as the location of controls, the format of the display, and the location of various switches and status indicators.

(3) <u>Device behavior knowledge</u> is the user's understanding of the relationships between the operation of various controls and the external behavior of the device.

(4) <u>How-it-works knowledge</u> is the user's understanding of the internal structures and functions of the device. Such knowledge is assumed to enable a user to generate explanations about the operating procedures and the behavior of the device, and about the device-dependent components of the task representation. [Ref 3:p. 367]

The combined total of knowledge in each of these categories determines the user's knowledge of the device and affects the user's ability to operate the device successfully.

3. Predicting Ease of Learning and Use

Kieras and Polson argue that, to make a system easy to operate, one needs to make the knowledge needed to operate a system as simple as possible, i.e., to minimize the cognitive complexity of the system. By extending the GOMS model to include production rules, quantitative predictions can be made on the ease of learning and use of a system. This is known as the cognitive complexity approach. [Ref 4:p. 1]

The GOMS model and the cognitive complexity approach both characterize the procedural knowledge the user must have in order to operate software, such as an operating system. GOMS describes the content and structure of this knowledge, while the cognitive complexity approach represents the amount of the knowledge. [Ref 4:p. 4]

Cognitive complexity models are descriptions of the required procedural knowledge expressed within a constrained production system architecture. The complexity of the rules and the number of rules needed to express the required knowledge are related to the complexity of the system and to the amount of knowledge that must be acquired in order to use it. [Ref 4:p. 5]

The cognitive complexity of an interface is represented through what Kieras and Polson refer to as a production system. For a job-task representation, a production system consists of (1) a working memory, (2) a collection of

production rules, and (3) an interpreter, according to this model. [Ref 4:p. 7]

The production system working memory contains representations of current goals and inputs from the environment and other information about the state of current and past actions [Ref 4:p. 6]. The production system model differentiates between two forms of knowledge: declarative knowledge and procedural knowledge. Declarative knowledge is defined by Kieras and Polson as the knowledge of facts. It is represented as a set of propositions organized as a semantic network. Procedural knowledge is defined as the knowledge of how to do things and consists of production rules. [Ref 3:p. 369]

A production rule is a condition and action pair of the form:

IF (condition) THEN (action).

The condition of a production rule is a statement about the external environment or the contents of working memory. If the condition of a specific rule is met, the action of the rule is executed. Production rules may be standardized in complexity by the use of style rules. Style rules are used to constrain a set of production rules to be uniform in size and amount of content. [Ref 4:p. 7]

The interpreter operates by alternating between "recognize" and "act" phases. During the recognize phase, the interpreter compares the conditions of all rules against the

contents of working memory to look for matches. During the act phase, the interpreter will execute all rules that have their conditions met. [Ref 4:p. 7]

According to Bovair and others, the standardized and constrained production rules that are used to carry out a given job or task can serve as a measure of the complexity of that job or task. Once the rules are identified, they can be counted. The resulting number may be used to generate quantitative predictions. An important hypothesis of the production system model is that the number of production rules correlates directly with complexity. More rules require more learning and also result in longer times to execute the specified operation [Ref 4:p. 6].

The Kieras and Polson cognitive complexity model provides a simple yet credible way to measure and evaluate cognitive and task complexity. This is the model that has been selected for use in this study.

C. TASK COMPLEXITY

As discussed by Wood [Ref 6:p. 60], task complexity has presented a major problem in accumulating data on task effects primarily because it has been difficult to define task complexity. The Task Complexity Model developed by Wood defines three components of task complexity: products, required acts, and information cues. Based on these

components, Wood defines three forms of task complexity: component, coordinative, and dynamic.

1. Task Complexity Components

Wood defines task products as entities that are created or produced by behaviors that can be observed and described independently of the behaviors or acts that produce them. In other words, products are the measurable results of acts. A product is an abstract quality of a task which is independent of the goals and expectations of individuals who perform or evaluate tasks. [Ref 6:p. 64]

Required acts include a pattern of behaviors with some purpose or direction. The direction of an act is usually implicit in the verb used (e.g., copy, rename, delete) and provides a focus toward the lower level of mental and physical activities that make up the act. It is this directional aspect which separates one act from another. [Ref 6:p. 64]

The direction of an act is the specific kind of activity or process carried out when an act is performed. Direction can be described independently of any individual who actually performs the act and any context in which the act is performed. This is a fundamental attribute of Wood's model. [Ref 6:p. 65]

Information cues are pieces of information about the attributes of stimulus objects. An individual bases the judgements he or she is required to make during the

performance of a task on the available information cues. [Ref 6:p. 65]

Required acts and information cues are important task inputs because they set upper limits on the knowledge, skills, and resources individuals need for successful task completion. Therefore, task complexity, which describes the relationships between task inputs, is an important determinant of human performance through the demands it places on the knowledge, skills, and resources of individual task performers. [Ref 6:p. 66]

2. Types of Task Complexity

Wood defines the component complexity of a task as a direct function of (1) the number of distinct acts that must be executed in the performance of the task and (2) the number of distinct information cues that must be procéssed in the performance of those acts. As the number of acts increase the knowledge and skill requirements for a task also increases, simply because there are more activities and events that an individual must be aware of and able to perform. [Ref 6:pp. 66-67]

Coordinative complexity refers to the nature of relationships between task inputs and task products. The form and strength of the relationships between information cues, acts, and products, as well as the sequencing of inputs, are all aspects of coordinative complexity. This includes timing,

frequency, intensity, and location requirements for performance of required acts. [Ref 6:pp. 68-70]

Dynamic complexity is due to the changes in the states of the world which have an effect on the relationships between task inputs and products. The parameter values for the relationships between task inputs and products are nonstationary. Changes in either the set of required acts and information cues or the relationships between inputs and products can create shifts in the knowledge or skills required for a task. [Ref 6:pp. 71-73]

Performance of a dynamically complex task requires knowledge about changes over time in the component and coordinative complexities of a task. Total task complexity is a linear combination of the component, coordinative, and dynamic complexities of a task. Variations in task complexity appear to produce changes in knowledge, skill, and effort requirements for successful task performance. These requirements or task demands depend on the combinations of component, coordinative, and dynamic complexities which make up total complexity. [Ref 6:pp. 73-74]

D. USER INTERFACES

Rasmussen describes a good user interface facility as one of the most important components of an information system. The value of a system is reduced greatly if it has not been provided a useful and flexible user interface [Ref 2:p. 152].

Much attention has been given to user interfaces in recent years and several distinct varieties of interfaces have been developed. Two of these, command language interfaces and direct manipulation, are the most common, and discussed below.

1. Command Language Interfaces

Command language interfaces require the user to communicate with the computer by typing a formal language with specific syntax. The user is required to learn and memorize the commands and the sequences needed for an operation. [Ref 7:p. 154]

As mentioned earlier, command language interfaces are preferred by expert users because all input is performed at the keyboard. The user can input and manipulate the interface without having to move from the keyboard to the mouse. This advantage of efficiency and flexibility on the part of the expert outweighs some advantages of direct manipulation type interfaces. [Ref 7:p. 155]

The major disadvantage of command language interfaces is that there are many sources for possible errors. According to Margono, these sources include the following:

1. Difficulty of memorizing commands. The following are common memory-related problems that result in errors.

    a. Confusion of the syntax of the language with English syntax.

    b. Inconsistency in the command language syntax.

c. Arbitrary syntax (e.g., commands consisting of punctuation marks).

d. Ease of making errors when typing the commands.

e. Mismatches between the user's intention in the task domain and the computer terminology or syntax. [Ref 7:p. 154]

2. Uncertainty of whether a certain command did what the user expected (i.e., the user may have to display a directory after performing a certain command to see if a file has been deleted, copied, etc.).

3. Inability to scroll command language-level directory lists backward and forward when searching for specific files. [Ref 7:p. 155]

Command language interfaces may give the user an impression of indirectness. The use of a specified language implies an intermediary system between the user and the computer. [Ref 7:p. 154]

2. Direct Manipulation Interface

According to Margono and others, most people form pictures in their minds of tasks to be performed. People who translate tasks into pictures may more easily understand, learn, and memorize when they can visualize objects and actions. In direct manipulation interfaces, the visual representation of the objects and actions should match the way people think about the problem. [Ref 7:p. 154]

Direct manipulation enables the user to control system activities by direct action on objects rather than by use of procedural language. The computer system provides feedback to the user concerning the user's actions on these objects. This

feedback is the cognitive material that is processed by the cognitive system. The use of direct manipulation is popular in software products because it gives the user a sense of satisfaction while interacting with the system. Additionally, the feeling of "directness" gives the user the impression that the computer and the interface are "invisible." [Ref 7:p. 154]

Icons are graphical representations of the objects or actions in a direct manipulation interface. When an icon is activated with the use of a mouse or other pointing device, the system performs an operation that can be associated with the user's visual representation of the operation. New users find it easier to recognize and select an icon rather than to learn the meaning of a set of command names and then recall the exact syntax of the command. Icons are therefore used to reduce the complexity of a system and make it easier to use. Interface systems using icons give an immediate positive impression that the system is easy to learn and use. [Ref 12:p. 105]

The major disadvantage of icon-based interfaces is the time required to move the hand from the keyboard to the mouse, locate the appropriate icon, use the mouse to move the cursor to the icon, and then select the icon. Experienced users sometimes find this process annoying and inefficient. [Ref 12:p. 108]

E. SUMMARY

To summarize, the complexity in human-computer interaction is a function of two components: task and user interface complexity. User interfaces can be classified based on their method of communicating between the user and the computer, two of which include command language and direct manipulation. There is reason to expect performance to be better using direct manipulation as opposed to command language interfaces. The question remains: What are the relative efficacies of direct manipulation and command language interfaces at multiple levels of task complexity?

III. METHODOLOGY

A. EXPERIMENTAL DESIGN

The purpose of this study was to examine the relative effectiveness of a DM interface (DMI) versus a CL interface (CLI) at different levels of task complexity. A mixed between-subjects/within-subjects design was used. As Table 1 indicates, the type of interface (DMI versus CLI) was operationalized as the between-subjects factor. Task complexity (simple versus complex) was operationalized as the within-subjects factor.

TABLE 1
EXPERIMENTAL DESIGN

	Interface Conditions	Task Conditions	
Between Subjects	CLI	Simple Task	Complex Task
	DMI	Simple Task	Complex Task

Within Subjects

1. Independent Variables

The independent variables for this study were (1) type of interface (CLI or DMI) and (2) task complexity. The two interfaces are described in detail in a later section. Task

complexity was calculated by determining the number of inputs and outputs required for each task. For the simple task set, the average number of inputs and outputs was determined to be five. For the complex task set, the average number was determined to be 24. Thus an average of approximately 80% more inputs and outputs was required for the complex task set over the simple task set.

2. Dependent Variables

The dependent variables for which performance was measured were (1) time to complete each task set, (2) number of errors that occurred during completion of the task set, and (3) the number of times on-line help was referenced. Each participant's thinking style also was determined. A Verbalizer-Visualizer Questionnaire (VVQ) test was used to measure the degree to which pictorial or verbal representations are normally stored and accessed by each subject as he or she carries out various tasks. The VVQ contains 15 questions, each pertaining to a visual or verbal way of thinking. Each subject was given a score on a scale of 1-15. A low score indicates strong verbalizing tendencies and a high score indicates strong visualizing tendencies. [Ref 13:pp. 109-124]

A user log was generated for each user as each task set was performed. For every operation, the total completion time, operation code, and description of the operation were

recorded in the user log. Additionally, the various errors that could occur automatically were coded to permit categorizing the errors for further analysis.

For the data gathering runs, time to complete the task set was calculated as the difference in minutes between the time the subject logged onto the system and the time that the last operation in a task set was completed. The time reported for the practice session also included the time required by the user to read the instructions and other information and time required to learn to use the interface.[1]

Errors included all task set operations that were performed incorrectly. Errors automatically recorded in the user log by the software were counted, along with others discovered during a manual review of the user log file. This manual review was necessary to determine whether all operations were performed and were performed correctly.

The number of times the user referred to the CLI or DMI help screens was automatically logged for each task set. This variable was considered a measure of the ease or difficulty of remembering how to use the interface, after training was completed.

---

[1] In some instances it was necessary to restart the practice session to allow the subject additional practice time. The total time for the practice session included the times for the initial practice session and the second practice session.

B. HYPOTHESES

A total of three hypotheses were tested for this study.

- **Hypothesis 1: Completion Time**

H_{1A} The time required to learn how to carry out the tasks and to complete the practice session will be significantly greater for the DM interface than the time required to learn and complete the practice session using the CL interface.

H_{1B} The time required to complete the simple task set will not be significantly different between the two interfaces.

H_{1C} The time required to complete the complex task set using the CLI will be significantly greater than the time required to complete the complex task set using the DMI.

- **Hypothesis 2: Number of Errors**

H_{2A} The number of errors that occur during the practice session will be significantly greater while using the DMI than the number of errors measured while using the CLI.

H_{2B} The number of errors that occur while completing the simple task set will not be significantly different between the two interfaces.

H_{2C} The number of errors that occur while completing the complex task set will be significantly greater while using the CLI than the number of errors measured while using the DMI.

- **Hypothesis 3: Number of Help References**

H_{3A} The number of times the help screen is referenced during the practice session will be significantly greater for the DMI condition than the number of times the help screen is referenced for the CLI condition.

H_{3B} The number of times the help screen is referenced during the simple task set will not be significantly different for the DMI condition than for the CLI condition.

H_{3C} The number of times the help screen is referenced during the complex task set will be significantly greater while using the CLI than the number of times the help screen is referenced while using the DMI.

C. SUBJECTS

Subjects for the experiment consisted of 27 active duty military officers enrolled in various master's degree programs at the Naval Postgraduate School, Monterey, California. Four subjects were female and 23 male. The subjects ranged in age from 27 to 38 years old, with a mean age of 31.5 years. The subjects were randomly divided into two groups. Of these, 13 used the CL interface and 14 the DM interface for the trials. The level of experience was controlled by selecting novice subjects; all subjects had little or no experience with computer operating systems.

D. APPARATUS

Both of the interface systems tested for the study were developed to interact with the Microsoft Disk Operating System (MS-DOS), and used the same hardware with the exception that a mouse was required for the DM interface and a keyboard for the CL interface. All trials, under all conditions, were conducted in a controlled environment on Unisys 386 personal computers with VGA color monitors and math co-processors. Task operations were conducted using floppy disks, referred to as the data disks. These data disks were used for both

interfaces. The subjects were at no time permitted access to the hard drive.

Subjects were seated comfortably in front of the terminals and were allowed to adjust viewing distance, keyboard placement, and the location of the mouse as desired. No more than five subjects were tested at any one time. The computer room was quiet, and distractions that might influence test outcomes were minimized.

E. TASKS

The tasks carried out by the subjects were the same for both interfaces. For the <u>simple task condition</u>, users manipulated 14 files stored in two directories. The <u>simple task set</u> consisted of the following.

1. Create a subdirectory called *plots* under the *root* (\) directory.

2. Create a new file called *twoplots.drs* in the newly-created directory called *plots*.

3. Delete the file called *project.bak* from the *project* directory.

4. Copy the *plot.drs* file that is contained in the *root* directory to *plot.bak* in the *root* directory.

5. Rename the file called *twoplots.drs* in the *plots* directory to *twoplot.bak* in the *plots* directory.

For the <u>complex task condition</u>, the complexity of the tasks was increased by the addition of directories and files, and by combining operations. A total of 15 subdirectories and 102 files were included for use in carrying out various tasks

in the complex task set. The complex task set included the following five tasks.

1. Copy the *package* file in the *business* directory to the *box* file in the *business* directory.

2. Rename the *papers* file in the *supplies* directory, which is a subdirectory of *business*, to *document* in the *supplies* directory.

3. Create a file called *car* in the *ground* directory, which is a subdirectory of *transpor*. Then sort the files in the *ground* directory by size.

4. Delete the *planes* directory. This requires deleting four files in the *jets* directory and two files in the *planes* directory, then deleting the *jets* directory and the *planes* directory.

5. Find the largest file of all the directories and rename the file to *large.fil*. This requires sorting 102 files in 15 different subdirectories, determining that the file named *ship* in the *transpor* directory is the largest file, and renaming it to *large.fil*.

F. COMMAND LANGUAGE INTERFACE

The same CL interface was used for the practice, simple and complex task sets. Subjects assigned to the CL prototype interface test condition received the Command Language Interface (CLI) instruction set (Appendix B). This instruction set included a brief introduction to computer operating systems and file management, and a detailed description of the CLI they would use.

The CLI was developed in the C programming language for the experiment. It was not designed to be a fully functional DOS shell. Instead, only the sets of tasks used for the

trials were supported (see Appendix C for a list of CLI commands available).

Figure 1 illustrates the basic window structure for the CLI. The interface includes three windows: (1) Directory window which displays the directory tree, (2) File window which displays the files stored in a specified directory identified in the label, and (3) Command window which displays user-input commands, as typed on the keyboard. The keyboard is the only input device used for the CLI condition.

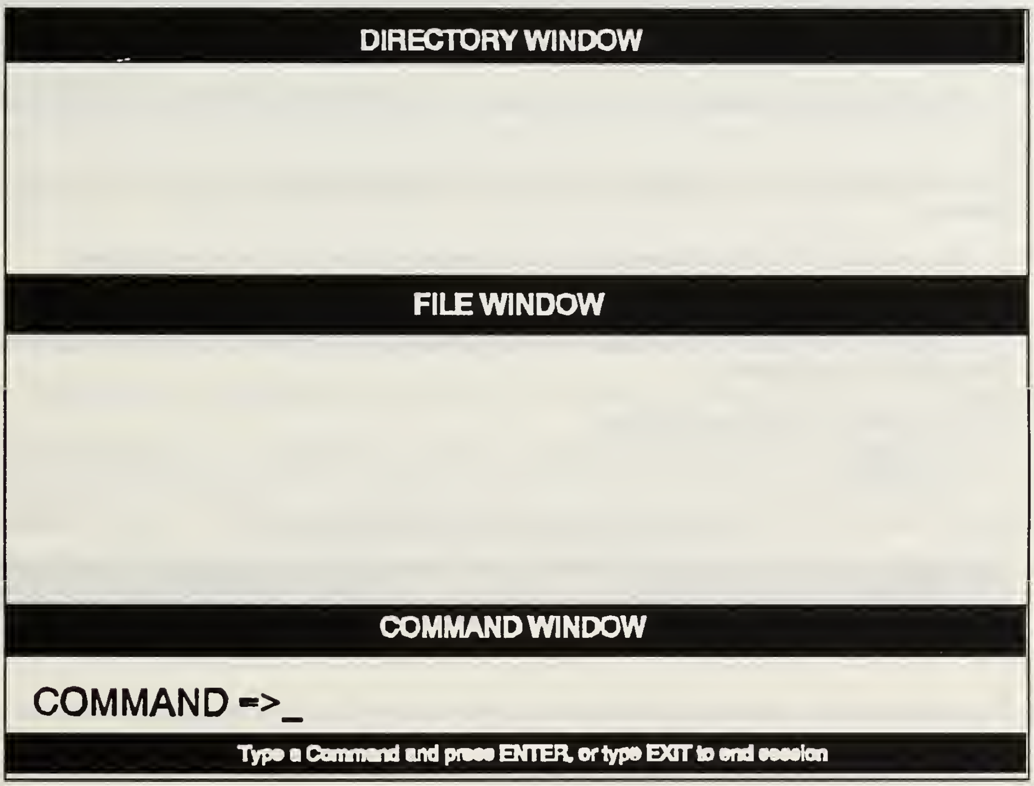

Figure 1 Command Language Interface Basic Window Structure.

Figure 2 shows the general Help screen that is displayed when the command **help** is entered at the command line in the Command window. If a portion of the screen is obstructed by the windows, the user is prompted to type "**M**" for more. The user can obtain a detailed description of each command by typing the words **help command**, followed by the name of the command for which information is desired. Figure 3 provides an example of the Help screen for the **createfile** command.

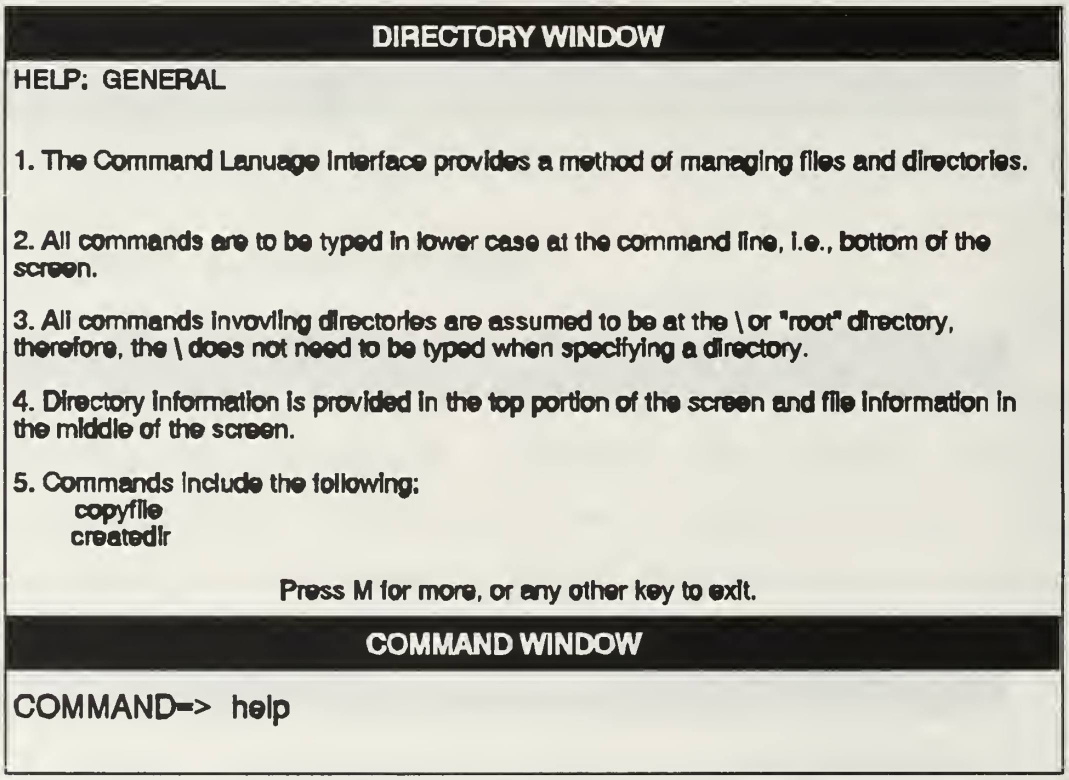

Figure 2 General CLI Help Screen.

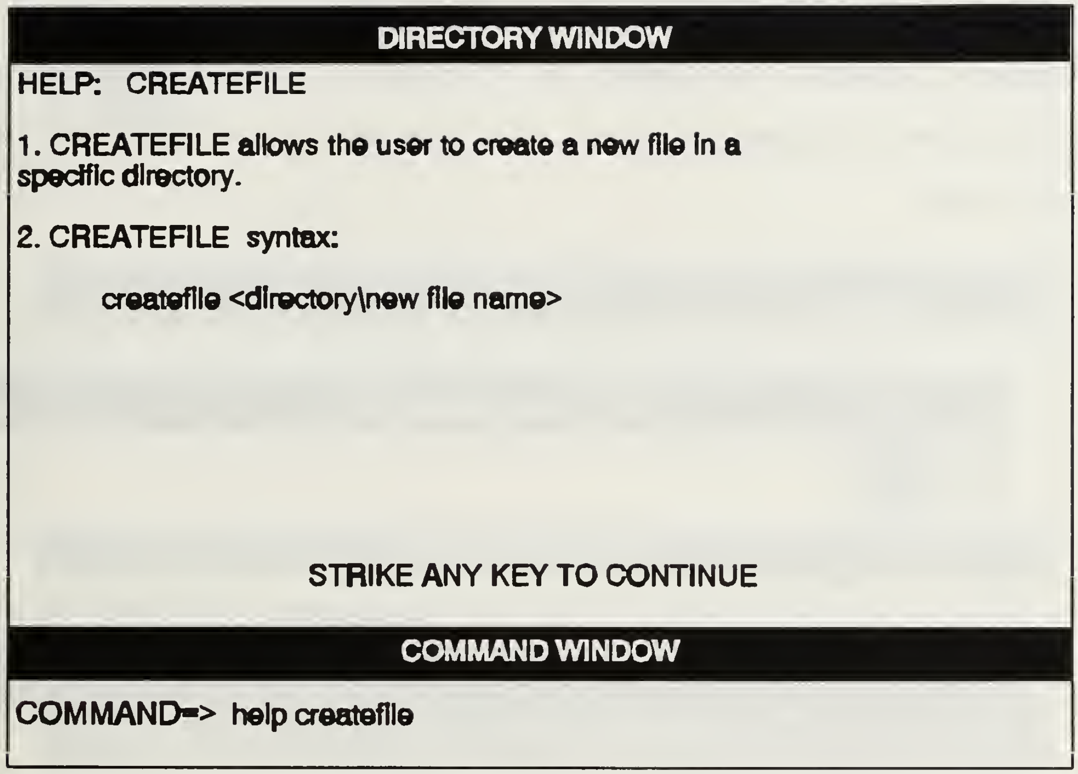

Figure 3 Help Screen for the Createfile Command.

Figure 4 illustrates the directory tree that is displayed when **dirtree** is typed on the keyboard and displayed at the command line. The Directory window then displays the names of all the directories contained on the data disk.

All commands are typed on the keyboard and displayed at the command line. Only one syntactical format is correct for each command. If the user types a command incorrectly, omits an argument, or specifies a non-existing file or directory, an

error message appears in reverse video at the base of the Command window, as shown in Figure 5. The error message indicates the type of error, but not the corrective action to be taken.

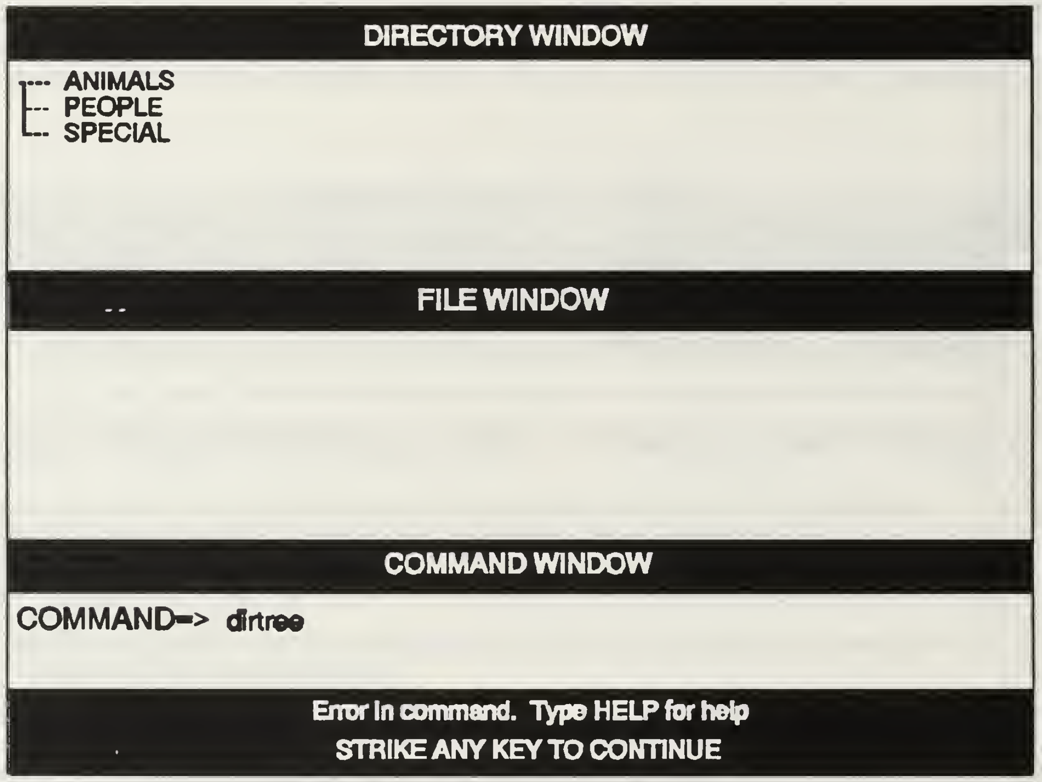

Figure 4 Directory Tree Screen for the CLI Condition.

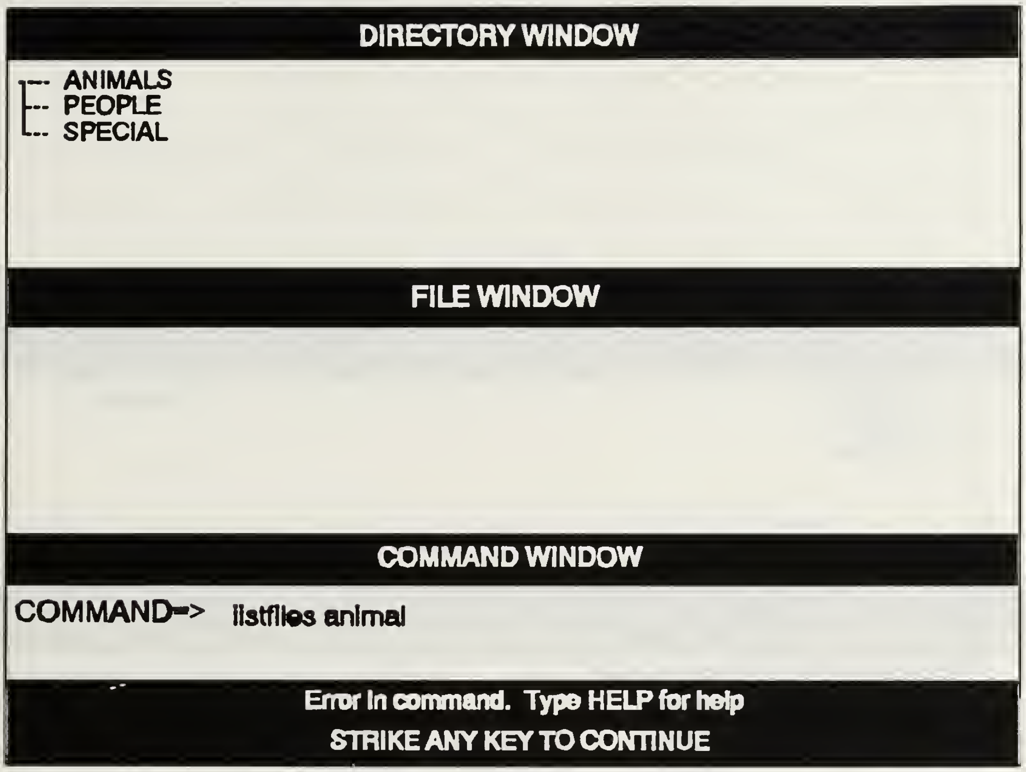

Figure 5 Screen with CLI Error Message.

1. Practice Session Screens

Figure 6 shows an example CLI screen used for the practice session. The user has given commands to display the directory tree (**dirtree**) and to list the files (**listfiles**) in the *animals* directory.

2. Screens Used for the Simple Task Set

Figure 7 shows an example CLI screen for the simple task set. The user has displayed the directory tree (**dirtree**) and listed the files in the *project* directory (**listfiles project**).

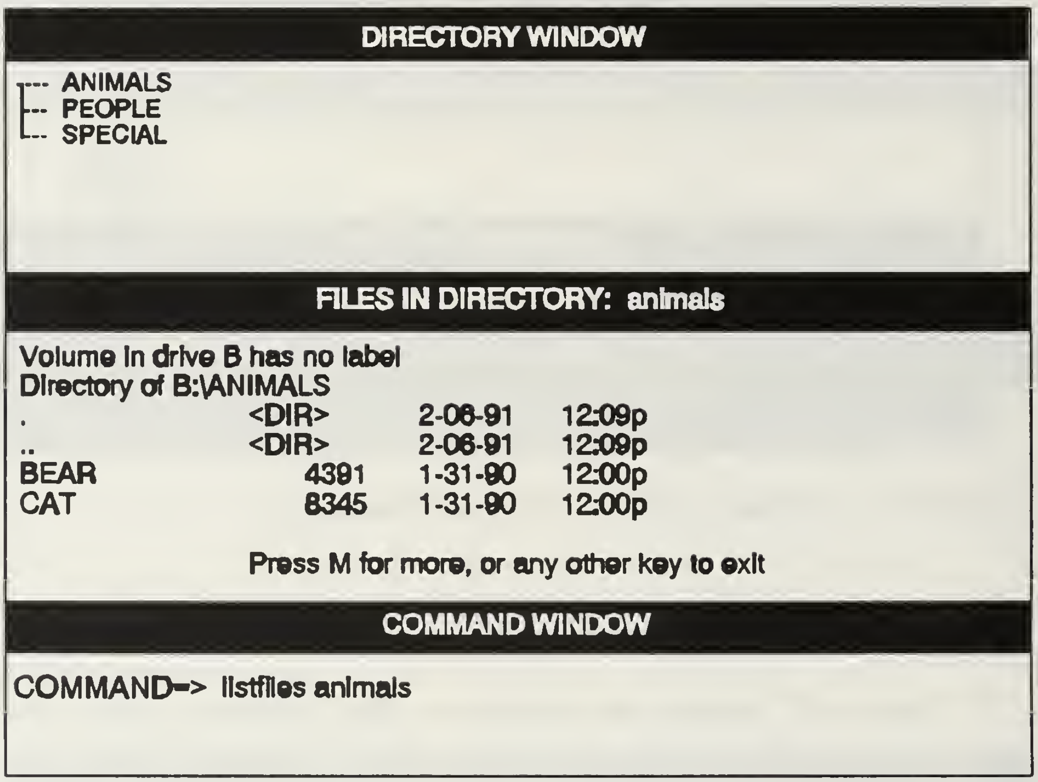

Figure 6 Practice Session Screen for the CLI Condition.

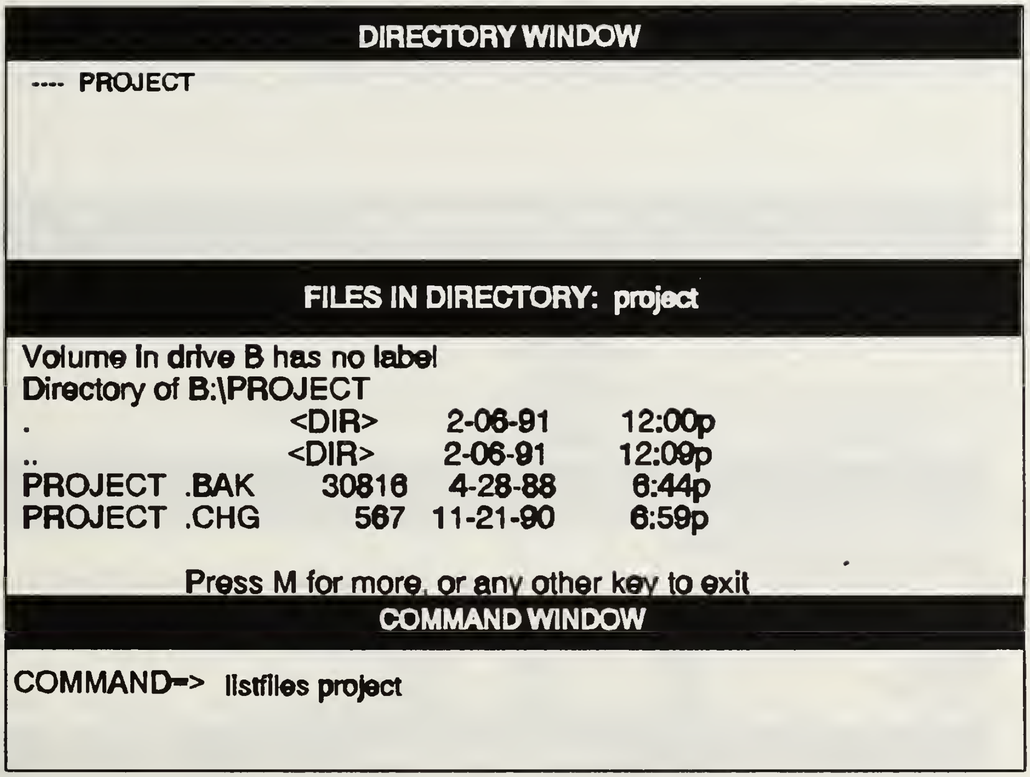

Figure 7 CLI Screen for Simple Task Set.

3. Screens Used for the Complex Task Set

The same interface with the same set of functions was used for the complex task set. Figure 8 provides an example of CLI screen for the complex task set. The user has displayed the directory tree (**dirtree**) and listed the files in the *business* directory (**listfiles business**).

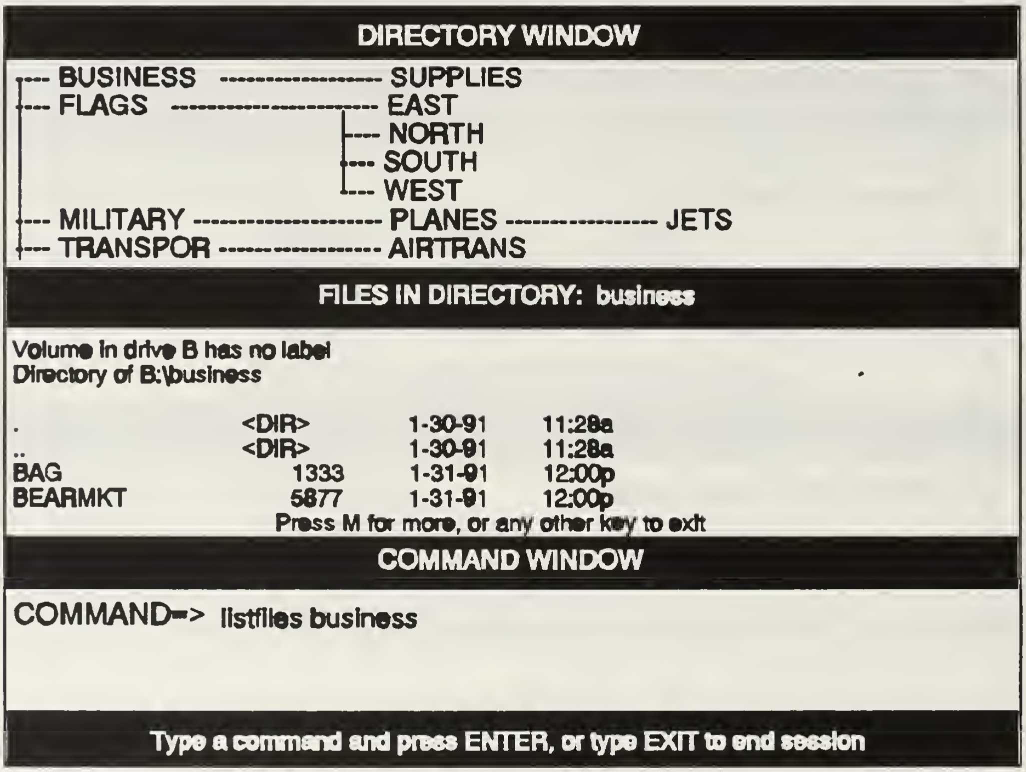

Figure 8 CLI Screen for Complex Task Set.

G. DIRECT MANIPULATION INTERFACE

The same DM interface was used for all three task sets. Subjects assigned to the DM prototype interface test condition received the Direct Manipulation Interface (DMI) instruction set (Appendix D). This instruction set included the same description of operating systems and file management as the CLI instructions, along with a detailed description of the DMI.

The DMI was programmed with the object-oriented programming language, **Smalltalk**, for the purpose of this experiment. Like the CLI, only the task sets used for the trials were supported by this interface (see Appendix C for a list of DMI commands available).

Figure 9 illustrates the basic structure for the DMI. The interface includes five windows: (1) Directory window which displays the hierarchical directory structure, (2) File window which displays the files listed in a specified directory, (3) File Sort window which allows the user to sort files by name, date of creation, or size, (4) New Name window which contains all allowed names needed for new files, renamed files, and new directories, and (5) Icon window which contains all the icons used for tasks operations. Figure 10 shows an example of an actual DMI screen. Although a keyboard was present, the mouse is the only input device used for the DMI condition.

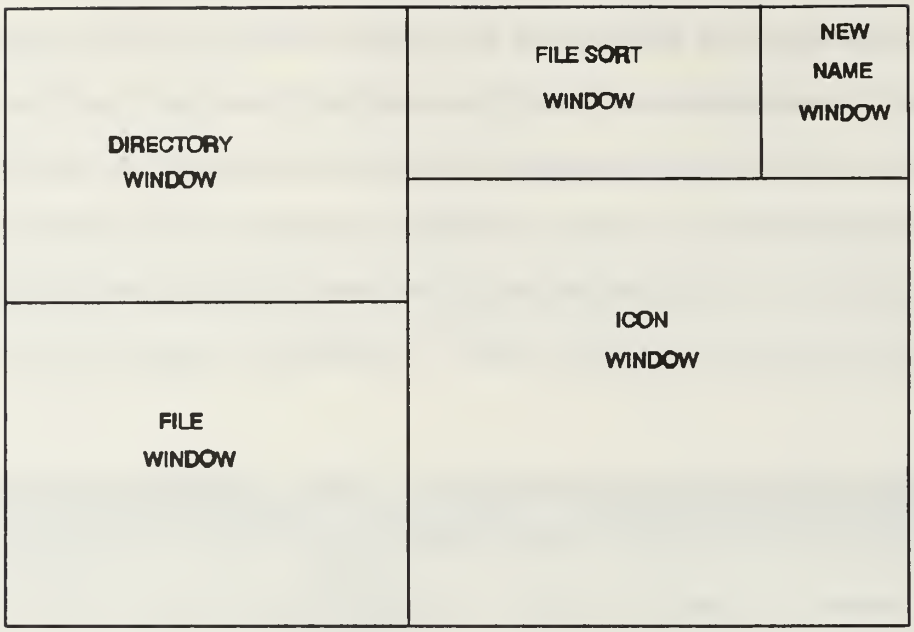

Figure 9 Direct Manipulation Interface Basic Window Structure.

Figure 11 shows the general Help window that is displayed on the screen when the **help** icon is selected. The user selects the name of the item in question by placing the mouse cursor over that name and pressing the left mouse button. The right side of the Help window then will display the step-by-step procedures for the specific operation. Figure 12 is an example of the Help window with the **Create Directory** option selected.

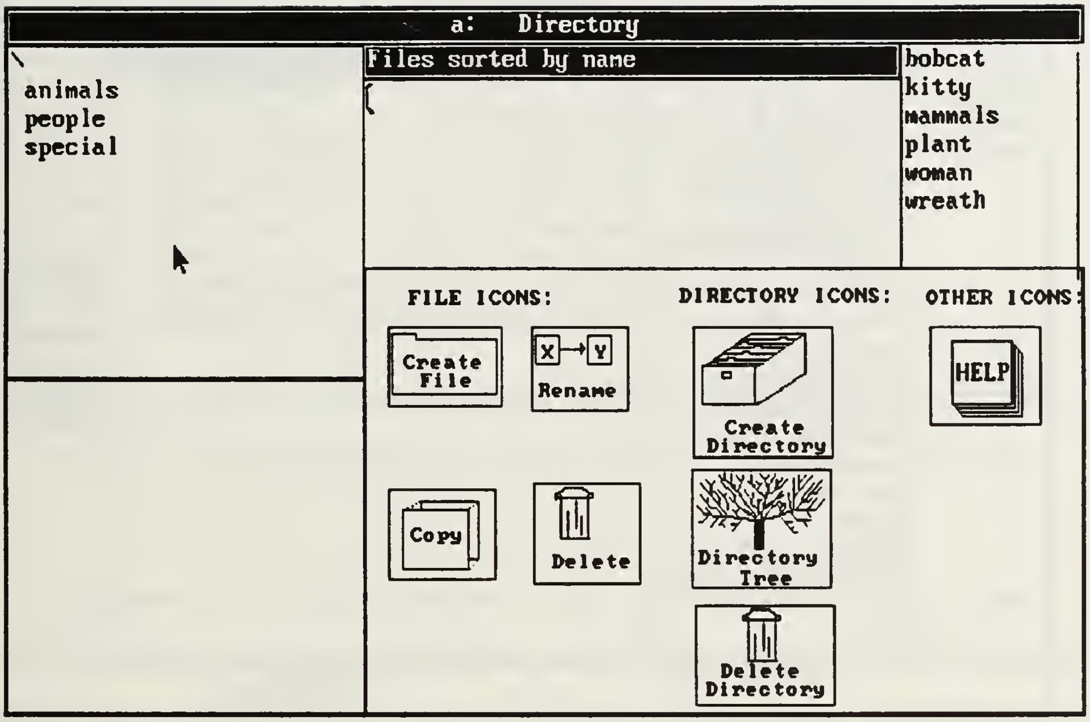

Figure 10 Direct Manipulation Basic Window Structure.

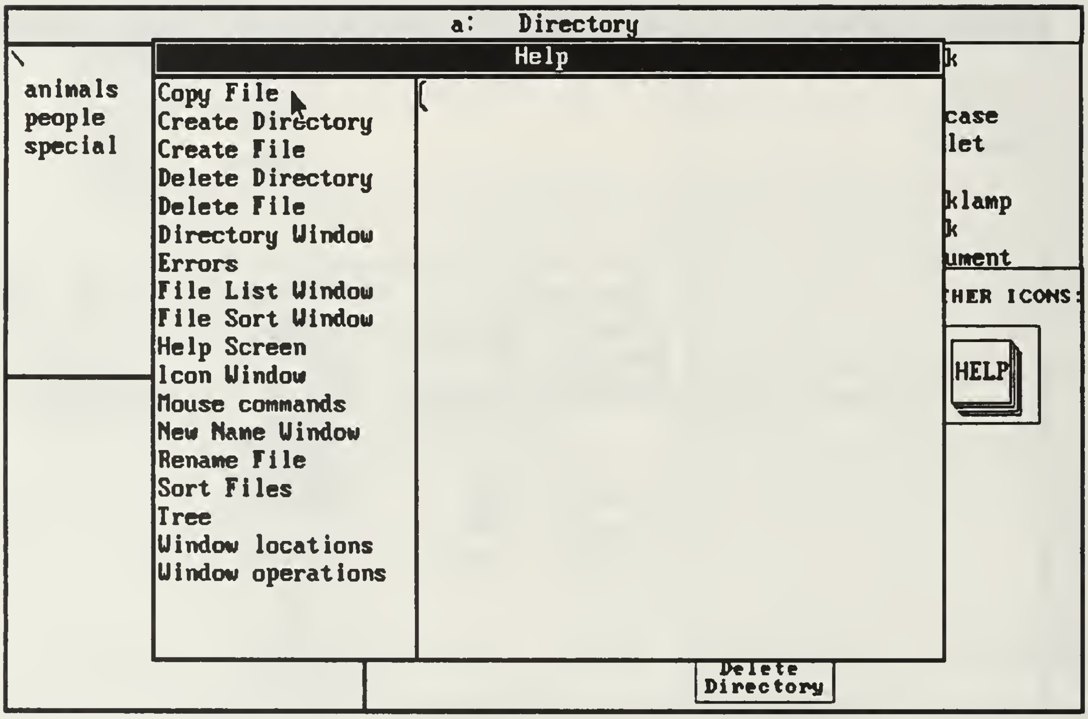

Figure 11 General DMI Help Window.

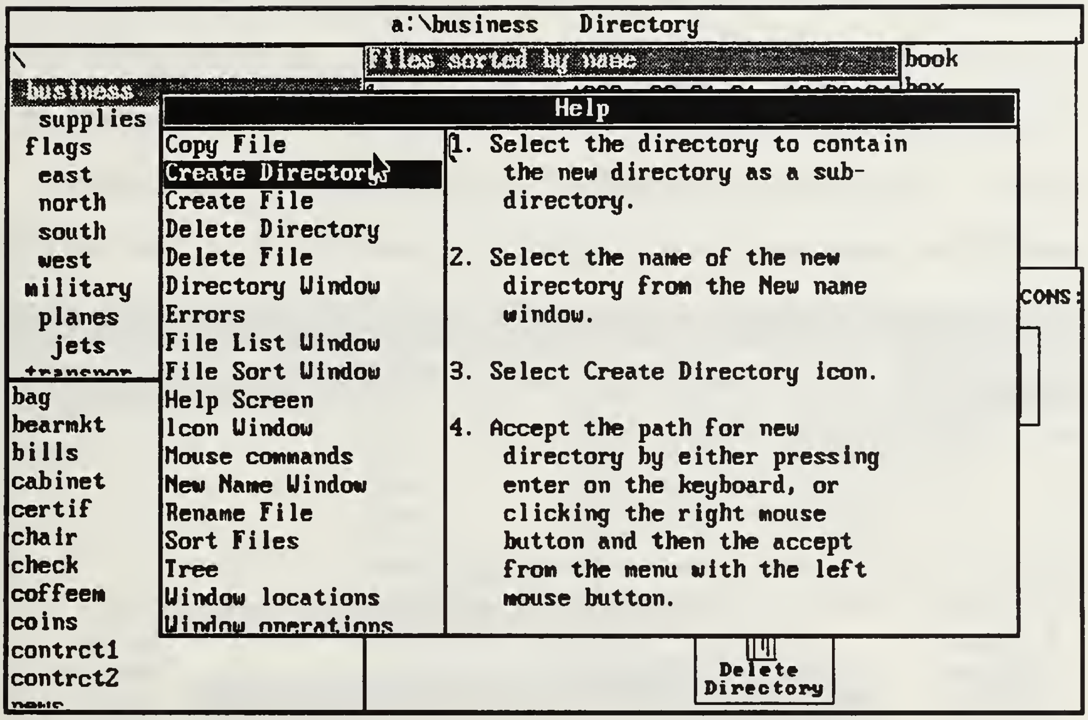

Figure 12 Help Window for the Create Directory Option.

Figure 13 illustrates the directory tree that is displayed when **directory tree** icon is selected. The Directory Tree window then displays the names of all the subdirectories contained on the data disk.

When the user performs an operation properly on a file or directory a Prompter window appears allowing the user to accept or cancel the operation, as shown in Figure 14. If insufficient information is provided for an operation, a Prompter window appears containing a simple error message, illustrated in Figure 15. As with the CLI, error messages indicate what has been done incorrectly, but not the

corrective action to be taken. If the user tries to perform an operation that is not allowed by DOS, an Error window with a short message in the label appears in the middle of the screen. This error message is generated by **Smalltalk**. The **Smalltalk** code generated during the execution of the operation is displayed in the lower portion of the window, as seen in Figure 16. This window is used to determine the reason for the error identified in the label.

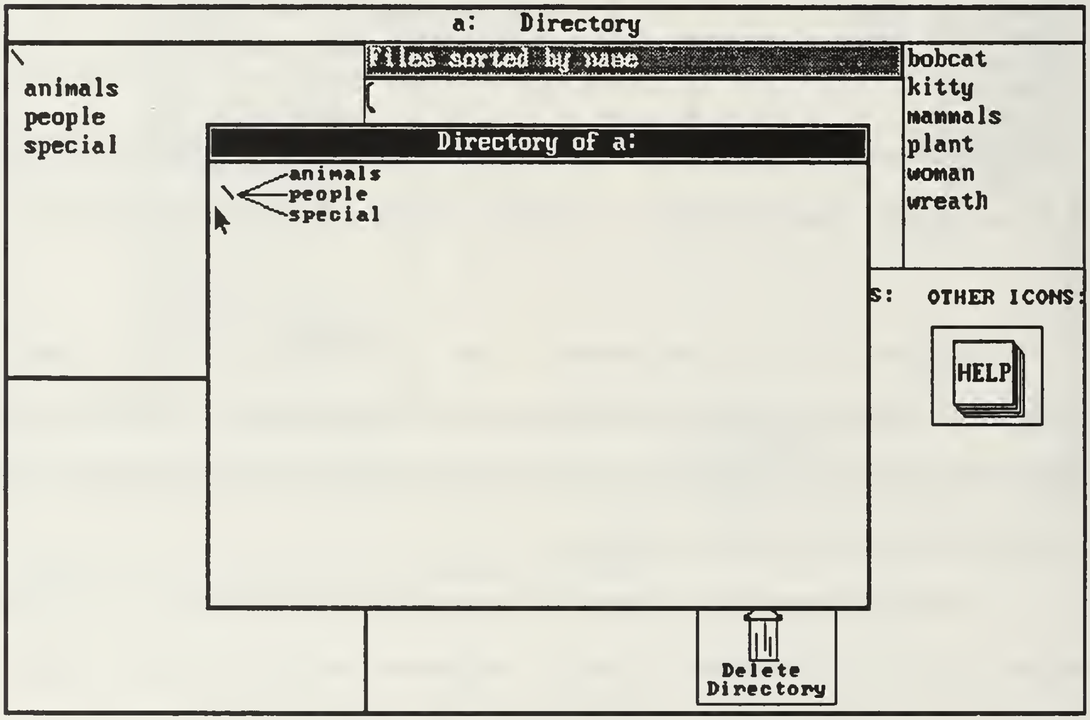

Figure 13 Directory Tree Window for the DMI Condition.

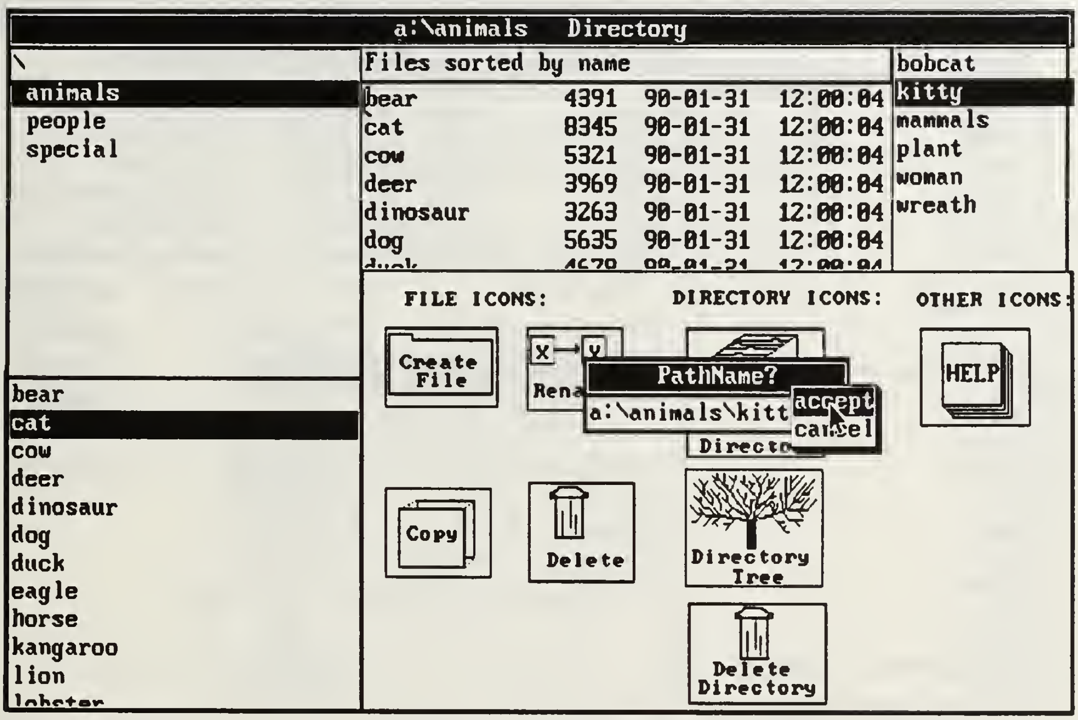

Figure 14 Prompter Window for the Pathname Option.

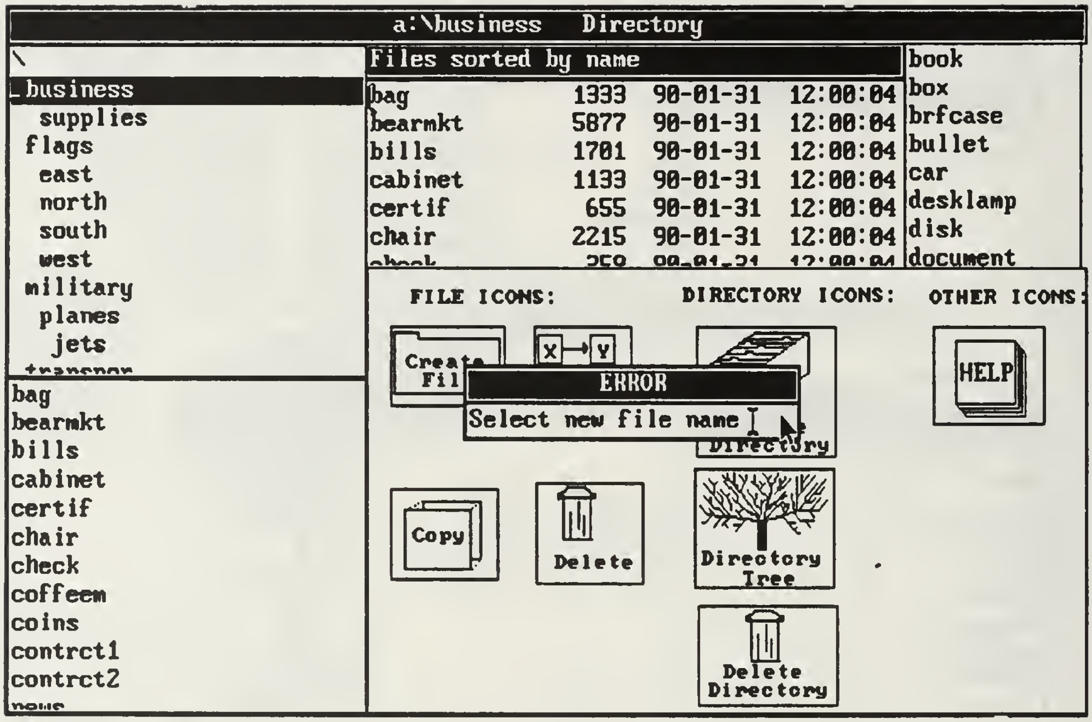

Figure 15 Prompter Window with Simple DMI Error Message.

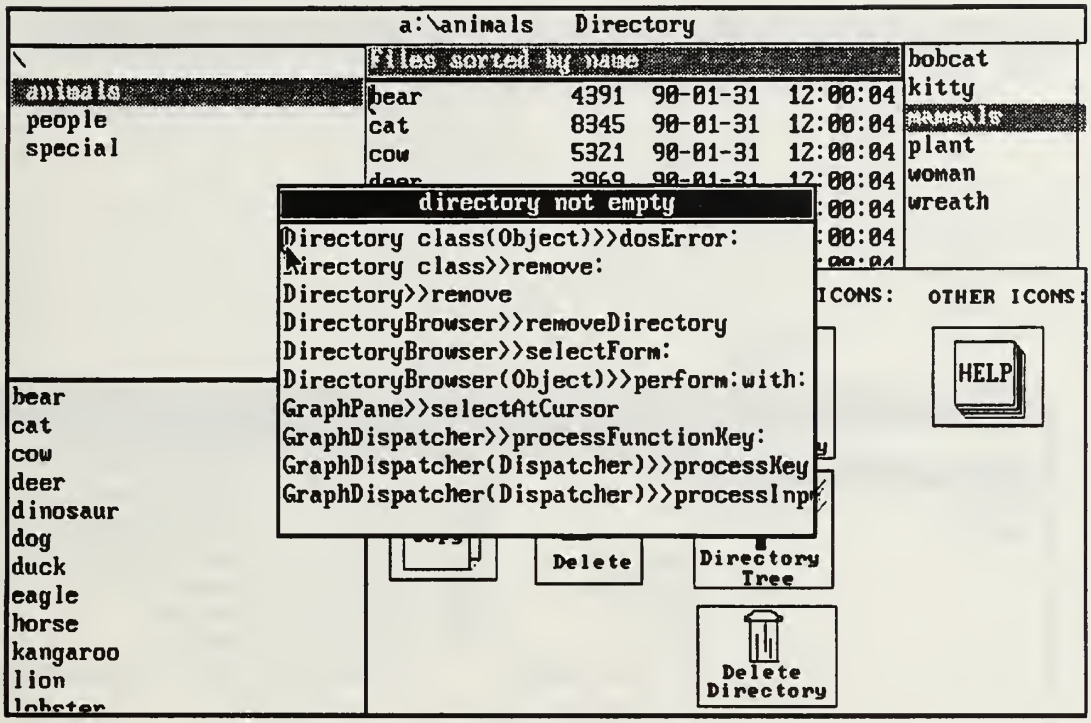

Figure 16 Smalltalk Error Window.

1. Practice Session Screens

Figure 17 shows an example DMI screen used for the practice session. The user has selected icons commanding that the *animals* directory be displayed, along with its files.

2. Screens Used for the Simple Task Set

Figure 18 shows an example DMI screen for the simple task set. The user has selected the *project* directory. The files are automatically displayed in the File window and the File Sort window.

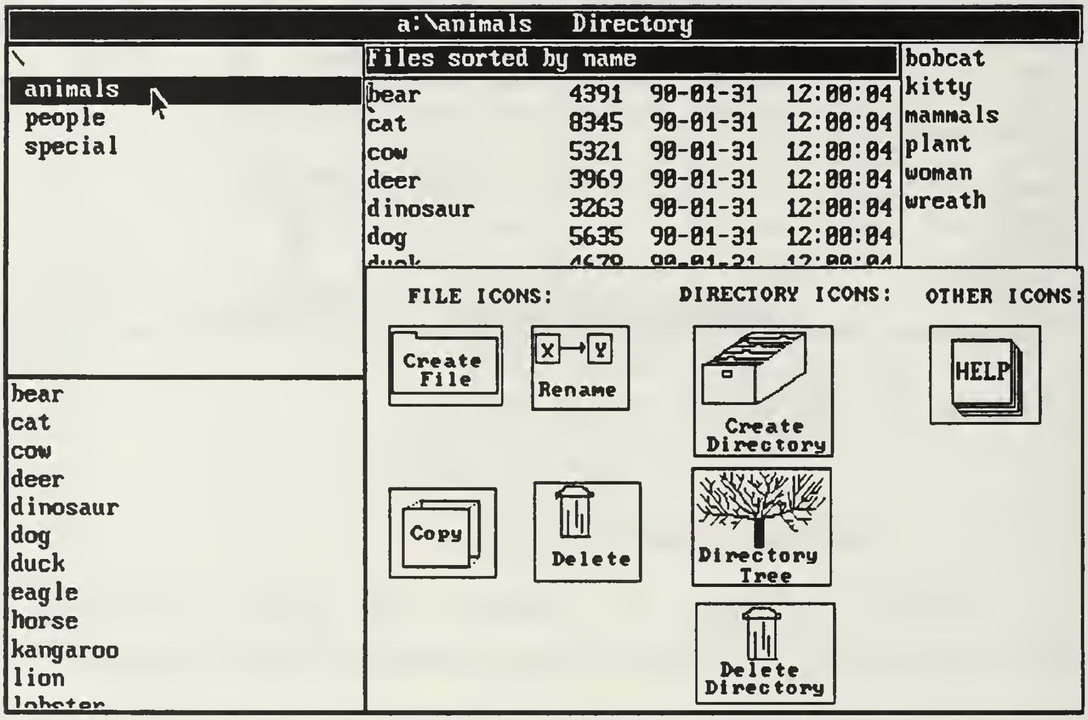

Figure 17 Practice Session Screen for the DMI Condition.

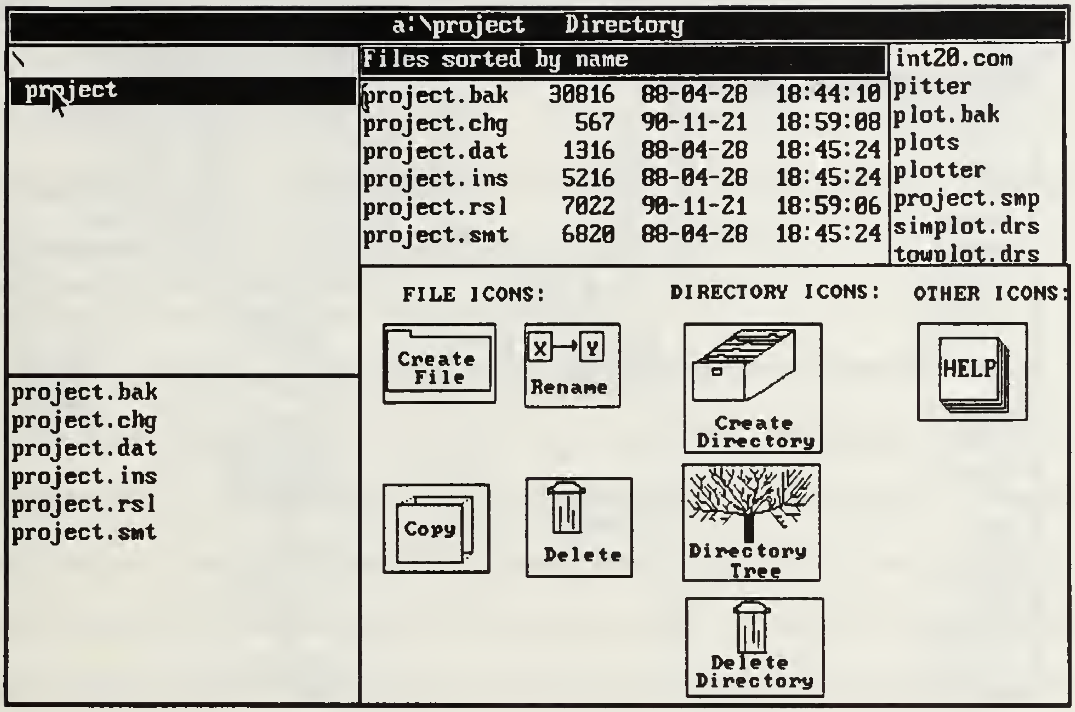

Figure 18 DMI Screen for Simple Task Set.

3. Screens Used for the Complex Task Set

The same interface with the same set of functions was used for the complex task set. Figure 19 shows an example DMI screen for the complex task set. The user has selected the *business* directory. The files are automatically displayed in the File window and the File Sort window.

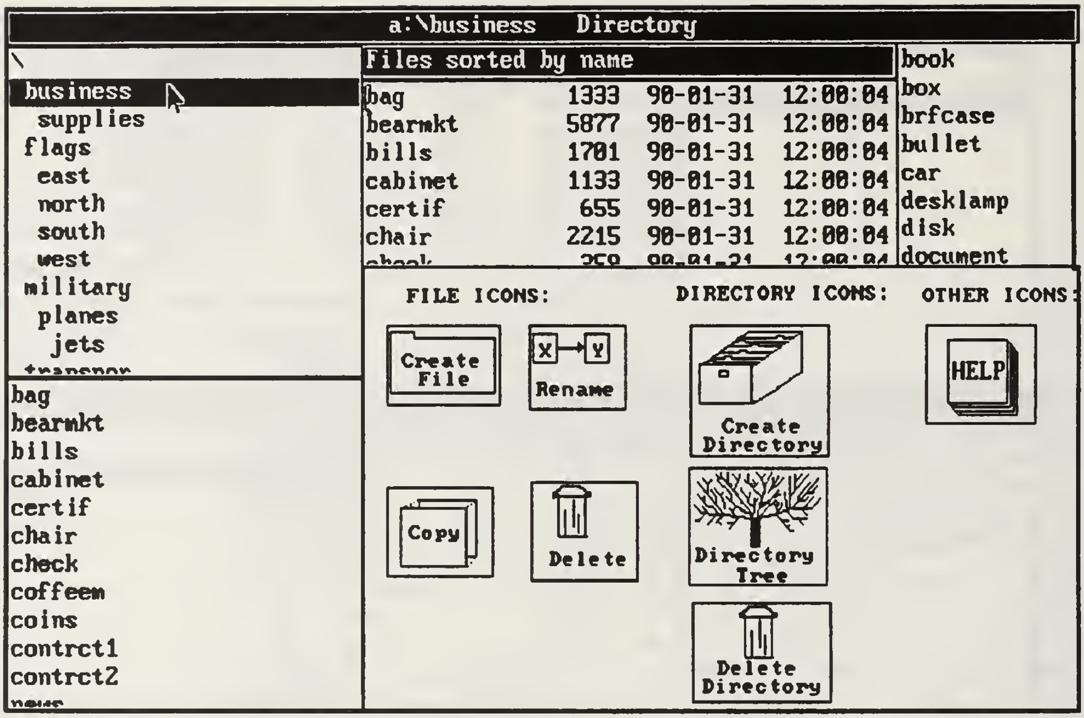

Figure 19 DMI Screen for Complex Task Set.

H. EXPERIMENTAL PROCEDURE

Subjects were brought into the computer room and told that they would be asked to carry out some tasks on a prototype operating system interface. Each subject then logged onto the Unisys 386 computer system by providing name and student mailbox number (SMC). The experimenter then provided each subject with an instruction set (Appendix B or D). The subject read a short introduction to the experiment and a privacy act statement. The VVQ was completed next, to determine thinking style (verbal or visual). The subjects

then read the information on file management, the operating system interface he or she would use, and the tasks. The two instruction sets (CLI and DMI) were identical except for the description of the interface and the practice session procedures. Information was provided only about the interface that would be used by that subject.

1. Training and Practice Session

After reading the instruction set the subjects were told to practice nine practice operations (described in the instruction set), repeating each as many times as necessary until it could be completed successfully. The nine operations were designed to familiarize the user with the interface and the operations to be used during the experiment. The practice session included the following file operations.

1. Call up on-line help
2. Create a file
3. Sort files
4. Delete a file
5. Copy a file and Rename a file.

Directory operations included:

1. Create a directory
2. Call up the directory tree
3. Delete a directory.

The user was not limited to these nine operations.

The file management and interface descriptions were available for reference during the practice session. After completing the practice session, each subject was asked if he or she felt comfortable with the interface, before continuing

with the experiment. If not, he or she was permitted to repeat the practice session. The average time spent in training and practice for CLI was approximately 9.9 minutes with a range of 6.4 to 17.5 minutes. Average time spent in training and practice for DMI was approximately 42.5 minutes with a range of 25.6 to 74.4 minutes.

Subjects proceeded at their own pace, and could call the experimenter if they encountered problems. The experimenter would intervene when called on for assistance by the subject, or when the subject had spent at least 5 minutes on a problem and was making no apparent progress in resolving it. An attempt was made to keep interventions to a minimum.

2. Data Collection Sessions

After completing the practice session successfully, the subject was instructed to notify the experimenter before proceeding. The experimenter then replaced the instruction set with information about the simple task set that was to be completed next. The complete file system was changed at this time to include a new set of directories and files, as required for the data trials. The subjects were instructed and encouraged to use the on-line help and directory tree as needed. Error messages also were provided. The error messages on both interfaces described what caused the error, but not necessarily how to correct the error.

After completing the simple task set, the experimenter was again notified so the computer files and interface could be prepared for the complex task set. A new directory and file structure was installed and the subject was instructed to proceed with the set of five complex tasks.

3. Post-Test Data Collection

After completing the complex task set, each subject answered a series of questions to obtain their opinions about the operating system interface used in the experiment. Questions also were asked about operating system experience, and experience with typing (CLI) or a mouse (DMI). Some personal data also was collected. This questionnaire is included in Appendices B and D.

Each subject rated the ease of use of the interface he or she had used on a scale of 1-9 with 1 being "not at all easy" and 9 being "very easy." The ease of learning the interface also was ranked on a scale of 1-9 with 1 being "not at all easy" and 9 being "very easy." The helpfulness of the error messages were ranked on a scale of 1-9 with 1 being "not at all helpful" and 9 being "very helpful." The subjects were asked to rank their progress on the interface on a scale of 1-9 with 1 being "not at all rapid" and 9 being "very rapid."

The subject was asked to predict his or her percent accuracy and to estimate the number of errors that occurred. Question 6 asked if at any time during the experiment the

interface appeared so difficult that he or she was ready to abandon the experiment.

The subject was also asked to rank his/her experience with computers prior to the experiment. This was also ranked on a scale of 1-9 with 1 being "no experience" and 9 being "very experienced." If the subject reported an experience level of 2 or greater on this question, he or she was asked to report the type of operating system experience (DOS, UNIX, Apple/MacIntosh, VAX, or other). This information was used to determine the subject's level of procedural knowledge concerning operating systems. To confirm the user's level of experience with operating systems, an additional nine questions were asked to test objectively the user's level of capability.

The subject was then asked to rank typing experience for the CLI or mouse experience with DMI on a scale of 1-9 with 1 being "no experience" and 9 being "very experienced." They concluded by providing demographic data including age, sex, undergraduate degree, NPS Curriculum, and years of computer experience.

I. DATA COLLECTION PROCEDURE

A user log was maintained for each subject, automatically recording total response time and actions performed by the subject for each task session. The user log was then converted to summarize all times required to complete each

task set, operations completed, and errors that occurred. Appendix E is an example of a user log for a CLI trial and Appendix F is a sample of a DMI trial user log.

All possible operations and errors were defined and coded for both interfaces prior to the experimental runs. Appendix G and H summarize the codes for CLI and DMI respectively. If an operation was performed correctly (i.e., there were no typing errors in the command for a CLI trial and all required steps were completed for a DMI trial), the response time and type of operation (including file and directory names) were recorded. If an error occurred, the time it occurred, operation code, and a description of the error were recorded. This log permitted tracking the user's progress and verifying all operations.

In a few instances it was possible for the subject to perform the task incorrectly, yet have it recorded as a correct operation (e.g., misspell the new file name or create a file in the wrong directory). For this reason all user logs were reviewed manually. Errors noted during the manual review were included in the analysis and classified as missing or incorrect operations. Additionally, unnecessary operations were noted.

IV. RESULTS

A. DATA ANALYSIS

The data was analyzed using a multivariate analysis of variance for repeated measures to determine the interrelationships of the dependent variables: task completion time, number of errors, and number of times help screens were referred to. The model for analysis was derived from Winer [Ref 14:pp. 630-633]. Additionally, a univariate analysis was conducted on each dependent variable.

After reviewing the data, it was determined that two of the DMI subject data sets were outliers, and were dropped from the data analysis. As a result of the unequal number of subjects evaluated under each condition (13 CLI and 12 DMI), the analysis was conducted using the General Linear Models Procedure of the SAS statistical software program [Ref 15:pp. 555-556].

B. EXPERIMENTAL RESULTS

Summary statistics (means and standard deviations) for each of the three dependent variables are given in Table 2. The mean values for these variables are plotted as a function of interfaces and task types in Figures 20 through 22. The graphs indicate that, as the task complexity increases,

individuals using a CL interface will require more time to complete their tasks, will make more errors, and will refer to on-line help screens more often. When a DM interface is used, time to complete simple tasks is approximately the same as the time for the CL interface and remains low for the complex task set. The number of errors begins and remains low, while references to the help function are high during practice, but drop sharply as the simple and complex tasks are carried out.

TABLE 2
SUMMARY OF RESULTS
MEANS
(STANDARD DEVIATIONS)

Dependent Variables	Interface Type					
	CLI			DMI		
	Practice Session	Simple Task	Complex Task	Practice Session	Simple Task	Complex Task
Completion Time, Min	9.93 (3.60)	9.78 (6.38)	43.56 (9.39)	42.53 (14.68)	6.82 (3.03)	16.82 (4.01)
Number of Errors	1.15 (1.14)	2.46 (2.54)	16.08 (6.40)	3.00 (2.30)	1.92 (1.50)	4.25 (2.80)
Number of Help References	1.61 (0.65)	3.77 (3.92)	9.92 (4.42)	6.33 (4.05)	1.08 (1.56)	2.42 (2.06)

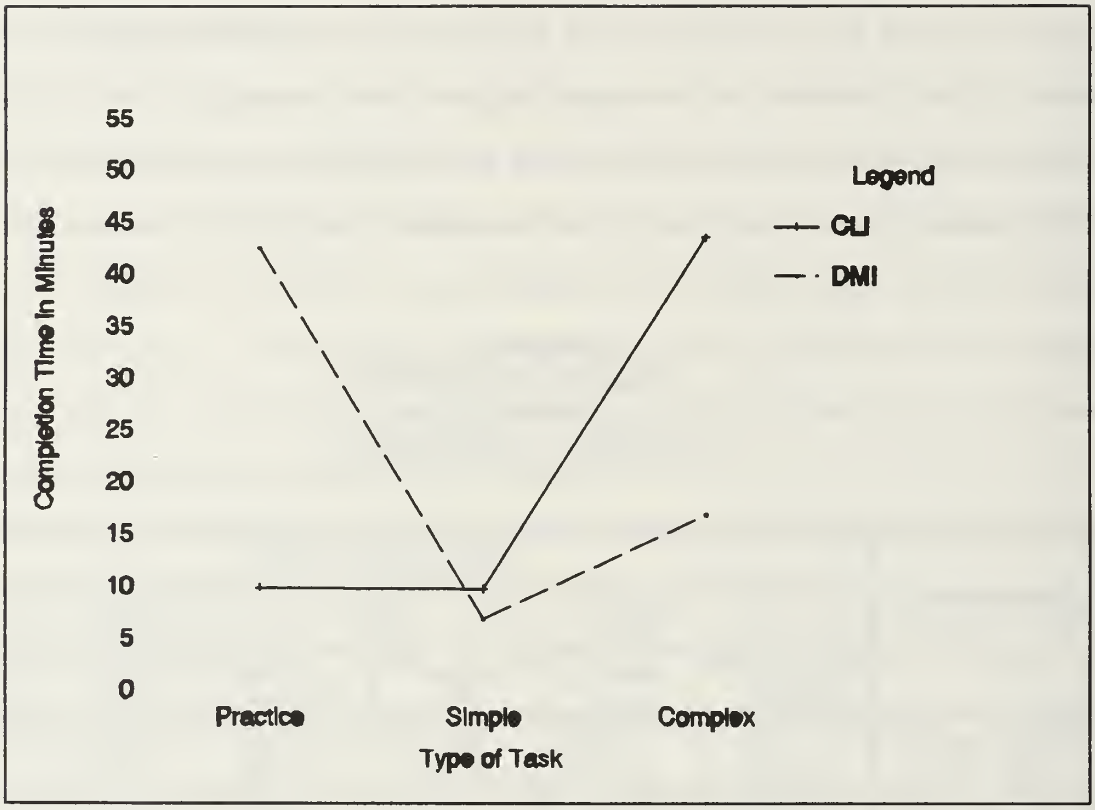

Figure 20 Mean Values for Task Completion Time, as a function of Interface Type and Task Type.

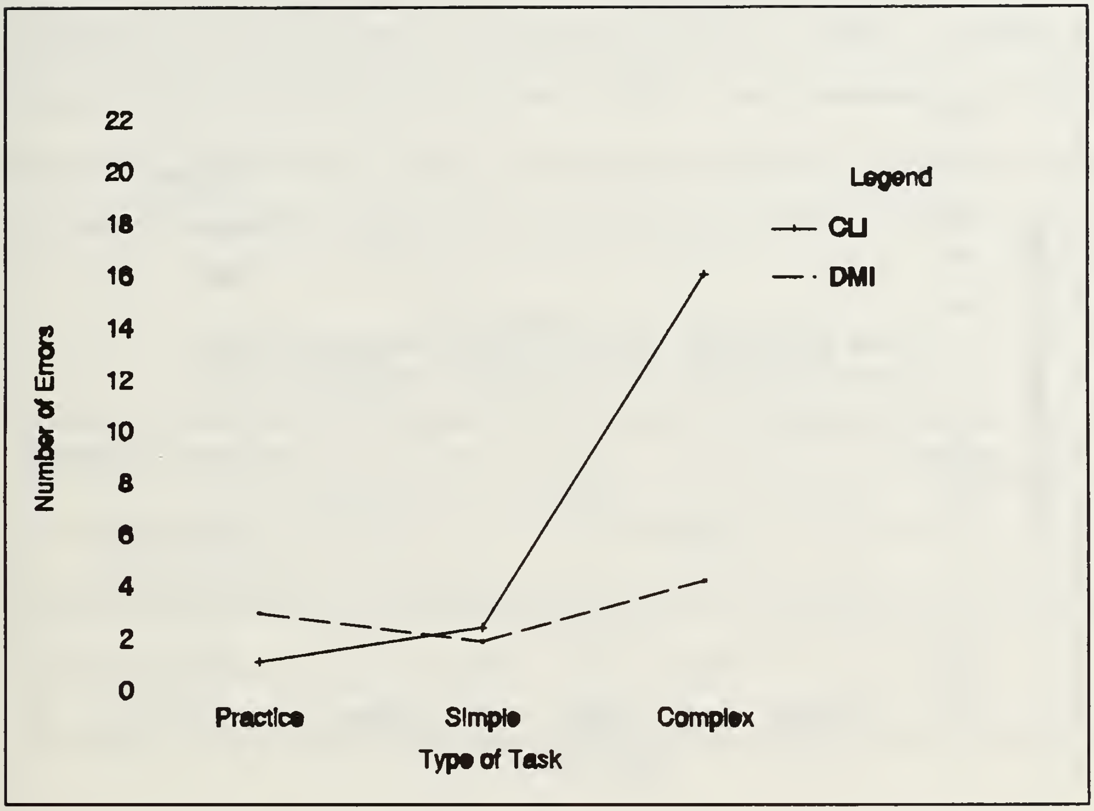

Figure 21 Mean Values for Number of Errors, as a Function of Interface Type and Task Type.

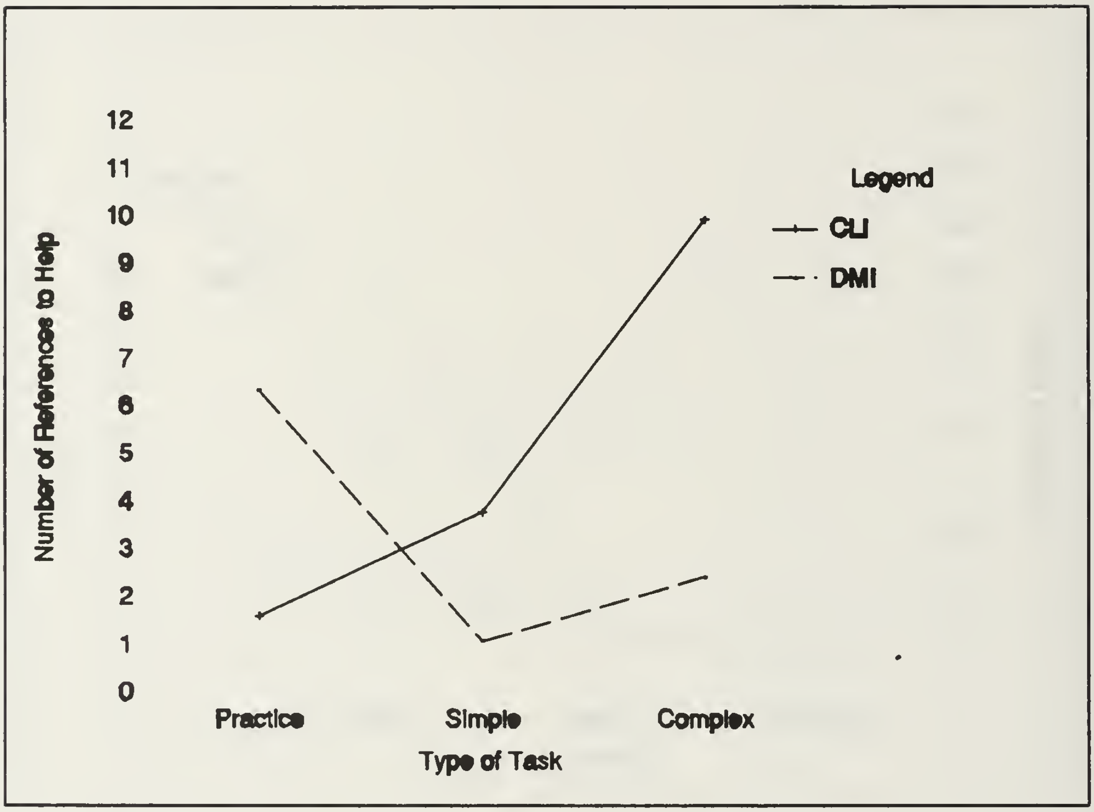

Figure 22 Mean Values for Number of Help References, as a Function of Interface Type and Task Type.

1. Task Completion Time

Table 3 summarizes the results of a multivariate analysis of variance for the dependent variable completion time. No significant difference was observed in task completion time as a function of the type of interface (CLI or DMI). There was a significant difference at the 0.0001 level between the tasks (simple or complex). The interaction between interface type and task completion time was significant at the 0.0001 level.

TABLE 3
MULTIVARIATE ANALYSIS OF VARIANCE FOR THE DEPENDENT VARIABLE COMPLETION TIME

Source of Variation	d.f.	F	SIG (F)
Interface Type	1	0.18	0.6748
Subjects within Groups	23		
Task Type	2, 22	214.30	0.0001
Task x Interface Type	2, 22	73.99	0.0001

Univariate analysis of variance tests were carried out to determine the direction of the interactive interface-task effects, and results were confirmed using Scheffe's test. As seen in Table 4, the time required to learn to use the DMI was significantly greater than the time to learn CLI at the 0.0001 level. This supports hypothesis H_{1A}, as discussed in Chapter III.

TABLE 4
UNIVARIATE ANALYSIS OF VARIANCE FOR COMPLETION TIME FOR PRACTICE SESSIONS

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	60.42	0.0001
Subject within group	23		

No significant difference in completion time was observed between the two interfaces for the simple task set (Table 5). This supports hypothesis H_{1B}.

TABLE 5
UNIVARIATE ANALYSIS OF VARIANCE FOR COMPLETION TIME FOR SIMPLE TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	1.83	0.1893
Subject within Group	23		

As Table 6 indicates, the time required to complete the complex task set using the CLI is significantly greater than task completion time using the DMI ($p<0.0001$). This finding supports hypothesis H_{1C}.

TABLE 6
UNIVARIATE ANALYSIS OF VARIANCE FOR COMPLETION TIME FOR COMPLEX TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	83.09	0.0001
Subject within Group	23		

2. Number of Errors

Table 7 summarizes results of a multivariate analysis of variance for the dependent variable <u>number of errors</u>. Performance results for the two interface types were significantly different at the 0.002 level. Errors also were significantly different as a function of task type ($p<0.0001$), and the interaction between interface type (CLI and DMI) and task type (simple and complex) also was significant at the 0.0001 level.

Univariate analysis of variance tests were also carried out on the results for the error variable, and confirmed using Scheffe's test. The number of errors that occurred during the practice session approached significance for subjects using the DMI, when compared with those using the CLI ($p<0.02$). This is shown in Table 8, and supports hypothesis H_{2A}.

TABLE 7
MULTIVARIATE ANALYSIS OF VARIANCE FOR THE DEPENDENT VARIABLE NUMBER OF ERRORS

Source of Variation	d.f.	F	SIG (F)
Interface Type	1	12.19	0.0020
Subjects within Group	23		
Task Type	2, 22	48.95	0.0001
Task x Interface Type	2, 22	25.66	0.0001

TABLE 8
UNIVARIATE ANALYSIS OF VARIANCE FOR NUMBER OF ERRORS, FOR PRACTICE SESSIONS

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	6.64	0.0169
Subject within Group	23		

No significant difference was observed in the number of errors generated by subjects using the two interfaces for the simple task set, as shown in Table 9. This supports hypothesis H_{2B}.

As seen in Table 10, the number of errors that occurred during completion of the complex task set was significantly greater for those using the CLI ($p<0.0001$). This finding supports hypothesis H_{2C}.

TABLE 9
UNIVARIATE ANALYSIS OF VARIANCE FOR NUMBER OF ERRORS, FOR SIMPLE TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	0.42	0.5248
Subject within Group	23		

TABLE 10
UNIVARIATE ANALYSIS OF VARIANCE FOR NUMBER OF ERRORS, FOR COMPLEX TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	34.78	0.0001
Subject within Group	23		

3. Help References

Table 11 summarizes the results of a multivariate analysis of variance for the dependent variable number of help references. Results approached significance for interface type ($p<0.05$). Statistical significance at the 0.0001 level was observed both for type of task and for the task type x interface type interaction.

TABLE 11
MULTIVARIATE ANALYSIS OF VARIANCE FOR THE NUMBER OF HELP REFERENCES

Source of Variation	d.f.	F	SIG (F)
Interface Type	1	4.51	0.0447
Subjects within Group	23		
Task Type	2, 22	17.14	0.0001
Task x Interface Type	2, 22	21.31	0.0001

Table 12 provides univariate analysis of variance results for the number of times that help was referenced during the practice session. As may be observed, differences in results between subjects using the two interface types was significant at the 0.0004 level. This supports hypothesis H_{3A}, that DMI help references will be greater than CLI help references.

TABLE 12
UNIVARIATE ANALYSIS OF VARIANCE FOR NUMBER OF HELP REFERENCES DURING PRACTICE SESSIONS

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	17.20	0.0004
Subject within Group	23		

Table 13 shows the univariate analysis results for the number of times that help was referenced during completion of the simple task set. Significance was approached (p<0.05), but not conclusive. Thus H_{3B} is supported.

TABLE 13
UNIVARIATE ANALYSIS OF VARIANCE FOR NUMBER OF HELP REFERENCES DURING SIMPLE TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	4.90	0.0370
Subject within Group	23		

Analysis of the number of times that help was referenced during completion of the complex task set is provided in Table 14. References by those using the CL interface were significantly greater than the number of referénces by DMI users (p<0.0001). This supports hypothesis H_{3C}.

TABLE 14
UNIVARIATE ANALYSIS OF VARIANCE FOR THE NUMBER OF HELP REFERENCES DURING COMPLEX TASK SET

Source of variation	d.f.	F	SIG (F)
Type of Interface	1	28.69	0.0001
Subject within Group	23		

Table 15 summarizes the posterior test results produced using Scheffe's test. The difference of the means were significantly different for all three dependent variables for the practice session and complex task set, and not significantly different for the simple task set. The same results were obtained using Tukey's posterior test.

TABLE 15
SUMMARY OF SCHEFFE'S POSTERIOR TEST

Dependent Variable	Practice	Simple	Complex
Completion Time	Significant	Not Significant	Significant
Number of Errors	Significant	Not Significant	Significant
Number of Help References	Significant	Significant	Significant

Legend:

Significant: Difference of means between two groups is significant at alpha=0.05.
Not Significant: Difference of means between two Groups is not significant at alpha=0.05.

C. SUBJECT RESPONSES

Participants in this study were asked to respond to the following questions, rating the interface used on a linear scale of 1 to 9.

1. How easy was this interface to use?

2. How easy was this interface to learn?

3. To what extent were the error messages helpful?

4. How rapidly did you progress through this experiment?

Results were averaged, and are provided as Figure 23. It should be noted that the subjects who used the DM interface perceived that interface to be easier to use and their progress more rapid than did those who used the CL interface. Table 16 summarizes the two-tailed T test which was conducted to determine the level of user agreement across the two groups. The t-value indicates that the subject responses are not significantly different between the two groups.

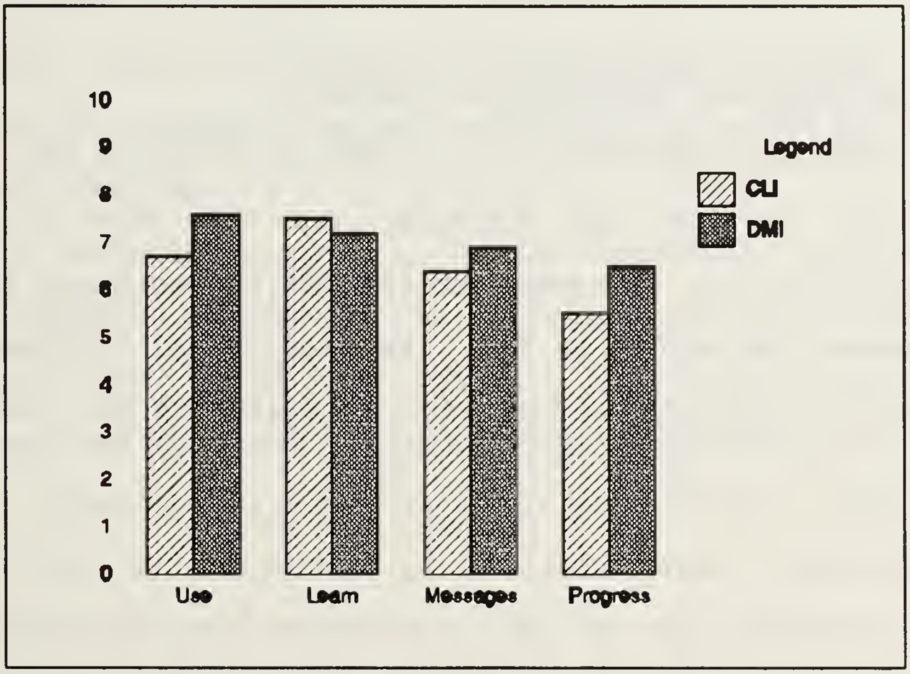

Figure 23 User Ratings of the CL and DM Interfaces, for Four Variables.

TABLE 16
RESULTS OF TWO-TAILED T-TEST
FOR SUBJECT RESPONSES
df=23, critical value = ± 2.069
at the 5% significance level

Question	Interface	Mean	Standard Deviation	t-value
Ease of Use	CLI	6.69	1.182	1.20
	DMI	7.58	2.275	
Ease of Learning	CLI	7.46	1.266	0.41
	DMI	7.17	2.125	
Error Messages Helpful	CLI	6.38	1.850	- 0.74
	DMI	6.38	1.782	
Rapid Progress	CLI	5.38	2.219	- 1.24
	DMI	6.50	2.276	

D. PROCESS AND MANIPULATION CHECKS

Table 17 summarizes the univariate analysis for the VVQ. The mean for the CLI subjects was 7.6 with a standard deviation of 1.8. The mean for the DMI subjects was 7.75 with a standard deviation of 2.4. No significant difference was observed in the VVQ for the subjects using the two interfaces.

TABLE 17
UNIVARIATE ANALYSIS OF VARIANCE FOR
VISUAL VERSUS VERBAL METHODS OF THINKING

Source of Variation	d.f.	F Value	SIG (F)
Type of Interface	1	0.03	0.8753
Subject within Group	23		

The number of subjects who contemplated abandoning the experiment was not different between the two interfaces. However, the point at which the subjects considered abandoning the experiment was meaningful. Three individuals using the CLI reported that they considered abandoning the experiment during the complex task session. All three reported that the steps needed to locate the largest file were too time consuming. This is supportive of hypothesis H_{1C}, that time to complete the complex task set using the CLI would be greater than with the DMI.

Two subjects assigned to the DMI condition reported that they considered abandoning the experiment during the practice session. Both found that learning the intricacies of the interface was difficult. This supports hypothesis H_{1A}, that the DMI interface would prove to be difficult to learn.

Of the 25 subjects, 14 indicated that they had some experience with DOS computer system, four had experience with Apple/MacIntosh systems, and five reported experience with "other" operating systems. A closer review of the "other"

category indicated that these subjects were uncertain as to the meaning of operating system, and mistakenly reported experience with DOS shells or various application programs.

Those with operating system experience were asked to complete a short quiz (contained in Appendices A and C). An average of less than 9% of the follow-on operating system questions were answered correctly by those who claimed to have some operating system experience. Thus, the assumption that the subjects had little or no operating system experience appeared to be valid.

The average age of subjects in the two groups was essentially the same (31.4 years for CLI and 31.5 years for DMI). Therefore, age was not a determining factor for performance on either interface.

V. CONCLUSIONS AND RECOMMENDATIONS

A. GOAL

The goal of this study was to determine if the user will benefit from the use of a DM interface over a CL interface for simple and complex tasks.

B. HYPOTHESES

The hypotheses listed below were supported or rejected using four statistical tests, multivariate analysis of variance, univariate analysis of variance, Scheffe's posterior test, and Tukey's posterior test:

- **Hypothesis 1: Completion Time**

H_{1A} The time required to learn how to carry out the tasks and to complete the practice session will be significantly greater for the DM interface than the time required to learn and complete the practice session using the CL interface. This hypothesis was <u>supported</u> by all four statistical tests.

H_{1B} The time required to complete the simple task set will not be significantly different between the two interfaces. This was <u>supported</u> by all four statistical tests.

H_{1C} The time required to complete the complex task set using the CLI will be significantly greater than the time required to complete the complex task set using the DMI. This was <u>supported</u> by all four statistical tests.

- **Hypothesis 2: Number of Errors**

H_{2A} The number of errors that occur during the practice session will be significantly greater while using the DMI than the number of errors measured while

using the CLI. This was <u>supported</u> by all four statistical tests.

H_{2B} The number of errors that occur while completing the simple task set will not be significantly different between the two interfaces. This was <u>supported</u> by all four statistical tests.

H_{2C} The number of errors that occur while completing the complex task set will be significantly greater while using the CLI than the number of errors measured while using the DMI. This was <u>supported</u> by all four statistical tests.

- **Hypothesis 3: Number of Help References**

H_{3A} The number of times the help screen is referenced during the practice session will be significantly greater for the DMI condition than the number of times the help screen is referenced for the CLI condition. This was <u>supported</u> by all four statistical tests.

H_{3B} The number of times the help screen is referenced during the simple task set will not be significantly different for the DMI condition than for the CLI condition. This was <u>supported</u> by all four statistical tests.

H_{3C} The number of times the help screen is referenced during the complex task set will be significantly greater while using the CLI than the number of times the help screen is referenced while using the DMI. This was <u>supported</u> by all four statistical tests.

C. CONCLUSIONS OF STUDY

This study has demonstrated that learning to use a DM interface is more time consuming than learning to use a CL interface. However, once it is learned, a novice can be productive in carrying out complex tasks, as well as, simple ones. The productivity of those using a CL interface appears to be reduced as the complexity of the task is increased.

That is, the time required to complete the task is increased, more errors are generated, and the number of references to help screens increases.

Users also perceive the DM interface to be harder to learn, yet easier to use. Additionally, the users of the DM interface perceive the error messages to be more helpful and progressed more rapidly through the experiment.

This study explored the relative benefits of DM interfaces and CL interfaces. Essentially, for simple tasks, neither interface has a clear advantage. However, as the complexity of the task is increased, DM interfaces appear to help novices to be more productive than do CL interfaces.

D. LIMITATIONS AND FURTHER RESEARCH

Several follow-on studies are suggested by the results provided here. First, to determine the level of task complexity at which a DM interface is more advantageous, a similar experiment could be conducted using several levels of task complexity instead of two. Second, user interface complexity (as discussed in Chapters I and II) should be considered. This form of complexity might be increased by providing more than one way to complete a given task.

Third, similar experiments could be conducted to determine relative CL and DM interface usefulness for other areas of computer applications such as word processing and data processing. The number of subjects tested should be

increased for such studies and should be representative of those likely to utilize such applications, i.e., clerical personnel.

Fourth, verbal protocol techniques could be used during conduct of similar interface experiments. These techniques aid in modeling the users' thought processes, providing useful information related to completing tasks.

Fifth, this study compared the productivity of persons using an icon-based DM interface to that of those using a CL interface. Other research could include comparisons of menu-based interfaces to icon or CL-based interfaces. This would determine which form of DM would be more productive.

Sixth, another area of research might include comparing novices and experts in either one or all of the suggested areas above. Such research would aid program developers in deciding which interface(s) to include in an application program for their typical users.

APPENDIX A

DEFINITIONS

Cognitive complexity: complexity dependent on the amount, content and structure of the knowledge required to operate a device or system [Ref 3:p. 364-365].

Cognitive Complexity approach: work based on the hypothesis that content, structure, and amount of knowledge required to perform a task on a system determines training time and productivity [Ref 2:p. 366].

Cognitive system: the human system that provides the ability to move symbolic information obtained from the sensory image stores to working memory, where it is combined with information previously stored in long-term memory [Ref 1:p. 24].

Command language (CL) interface: an interface that requires the user to communicate with the computer by typing a formal language with specific syntax [Ref 6:p. 154].

Complexity: the number of goals and processes that must be controlled, and the characteristics of the means available for this control [Ref 10:p. 24]; the level of complexity is determined by the difficulty of acquiring the new knowledge necessary to operate a device successfully [Ref 2:p. 366].

Component complexity of a task: complexity that is a direct function of (1) the number of distinct acts that must be executed in the performance of the task and (2) the number of distinct information cues that must be processed in the performance of those acts [Ref 6:p. 66-67].

Coordinative complexity: complexity due to the nature of relationships between task inputs and task products [Ref 6:p. 68].

Declarative knowledge: knowledge of facts [Ref 3:p. 369].

Device complexity: the level of complexity of the knowledge representations required to operate a given device [Ref 2: p. 367].

Device representation: the knowledge that a user has about the device itself [Ref 2:p. 366].

Direction: a specific kind of activity or process carried out when an act is performed [Ref 6:p. 65].

Direct manipulation (DM) interface: an interface that presents a set of visual representations on a display and provides a repertoire of manipulations that can be performed on any of them [Ref 11:p. 283].

Dynamic complexity: changes in the states of the world which have an effect on the relationships between task inputs and products [Ref 6:p. 71-73].

GOMS model: a descriptive model that describes a user's task representation using goals, operations, methods, and selection rules [Ref 1:p. 139-144].

Human-computer interface: the dialogue that occurs between a user and a computer to accomplish a specific task [Ref 1:p. 23].

Icons: graphical representations of the objects or actions that are part of a direct manipulation interface [Ref 12:p. 105].

Information cues: pieces of information about the attributes of stimulus objects [Ref 6:p. 65].

Motor system: the human system that carries out the responses of the perceptual and cognitive systems [Ref 1:p. 34].

Perceptual system: the human system that carries sensations of the physical world detected by the body's sensory systems into internal representations of the mind by means of integrated sensory systems [Ref 1:p. 25-34].

Production rule: a statement about the external environment or the contents of working memory [Ref 4:p. 7].

Production system: a formal notation used to represent the user's knowledge of the job-task environment, which is composed of production rules and working memory [Ref 2:p. 368].

Required acts: human acts that include a pattern of behaviors with some purpose or direction [Ref 6:p. 64].

Task complexity: the measurable complexity of the task that is being performed [Ref 6:p. 66].

Task products: entities that are created or produced by behaviors that can be observed and described independently of the behaviors or acts that produce them [Ref 6:p. 64].

Task representation: a user's knowledge of how to carry out a task using a given device [Ref 2:p. 366].

User interface complexity: the complexity of a device or system from the point of view of the user [Ref 2:p. 365].

Working memory: the part of memory that contains representations of current goals and inputs from the environment and other information about the state of current and past actions [Ref 4:p. 6].

APPENDIX B

COMMAND LANGUAGE INTERFACE

YOUR NAME: ______________________________

SMC NO: ______________________________

I. INTRODUCTION

The exercise you are about to participate in involves operating and evaluating a recently-developed computer file management system. You will be asked to read a short description of file management, followed by instructions for each file management operation. You will then practice each operation until you can perform the operation successfully. Upon completing all the operations successfully, you will do a series of tasks that involve using the operations you have learned.

PRIVACY ACT

The information accompanying this experiment will be used for data collection and correlation purposes only. Information provided is voluntary.

II. QUESTIONNAIRE

INSTRUCTIONS:

Please place a T(true) or F(false) next to each of the following statements as you feel it best applies to you.

____ 1. I enjoy doing work that requires the use of words.

____ 2. My daydreams are sometimes so vivid I feel as though I actually experience the scene.

____ 3. I enjoy learning new words.

____ 4. I can easily think of synonyms for words.

____ 5. My powers of imagination are higher than average.

____ 6. I seldom dream.

____ 7. I read rather slowly.

____ 8. I cannot generate a mental picture of a friend's face when I close my eyes.

____ 9. I don't believe that anyone can think in terms of mental pictures.

____ 10. I prefer to read instructions about how to do something rather than have someone show me.

____ 11. My dreams are sometimes vivid.

____ 12. I have better than average fluency in using words.

____ 13. My daydreams are rather indistinct and hazy.

____ 14. I spend very little time attempting to increase my vocabulary.

____ 15. My thinking often consists of mental pictures or images.

III. FILE MANAGEMENT SYSTEM

A. OPERATING SYSTEM

An **operating system** is the software program that makes the hardware useable. The operating system can accomplish many functions such as communicating between the user and the computer (known as the user interface), sharing hardware among users, allowing users to share data among themselves, and many other functions.

An operating system's primary duty is to manage various **files**. Before we go into the details of files, it is important to understand the structure and organization of files.

B. DIRECTORIES AND FILES

A **software program** usually consists of several files. These files work together to produce the program that the user sees and interacts with on the screen. A **directory** contains all of the files used for a given program. For instance, all the files that operate the popular word processing program, WordPerfect 5.1, might be located in a directory called WP51. It is also possible (and highly recommended) to have **subdirectories** within a directory to segregate your files further. For instance, in the WP51 directory you might have a subdirectory titled WORK for all the files relating to your work and one titled THESIS for all the files relating to your thesis.

Figure 1 shows the relationship of this WP51 directory to its subdirectories and files, and also its relationship with the top-level **Root** directory, often represented by a slash (\). Two other directories at the same level as WP51 (here called DOS and HG) also are shown, along with their files.

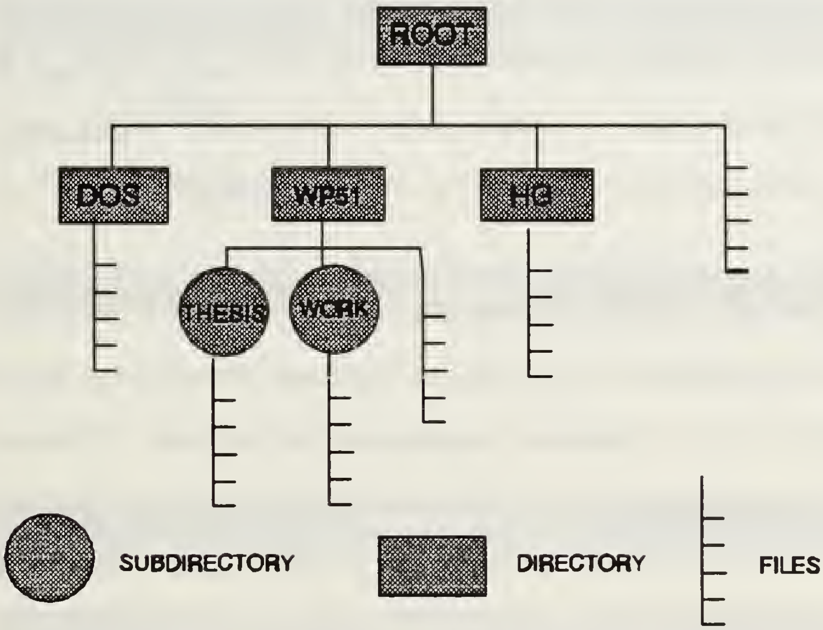

Figure 1. Relationships of the WP51 Directory with other Directories and Files.

It is often necessary to *copy* or *move* a file from one directory to another. The operating system must provide methods to move these files. For example, suppose you wrote a paper for a National Security class titled **The Middle East**

and stored this file in the WORK directory under WP51. Later in the year you decide to use this paper in your thesis and need to transfer the file to the THESIS subdirectory. Some form of *move* or *copy* command would allow you to transfer this file. Additionally, you may want to change the file name to **Chapter II**. Again the operating system interface must provide a means of doing so.

As mentioned earlier, directories contain related files. The operating system interface also must provide a means to **manage** directories. That is, commands such as *create directory* and *delete directory* are needed.

IV. OPERATING SYSTEM INTERFACE

The experimental operating system interface you will be working with is a **Command Language Interface**, Figure 2. It uses specific commands to perform an operation. The interface contains three windows: Directory, File, and Command windows.

A. WINDOWS

1. Directory Window

The Directory Window provides a directory tree of all the subdirectories contained in a specified directory. If insufficient space is provided in the window for the entire directory tree, the user will be prompted to press M for the remainder of the tree.

2. *File Window*

The File Window contains a listing of all the files in a specified directory. The name of the directory appears in the label for the File Window. If insufficient space is provided in the window for all the files, the user will be prompted to press M for additional files.

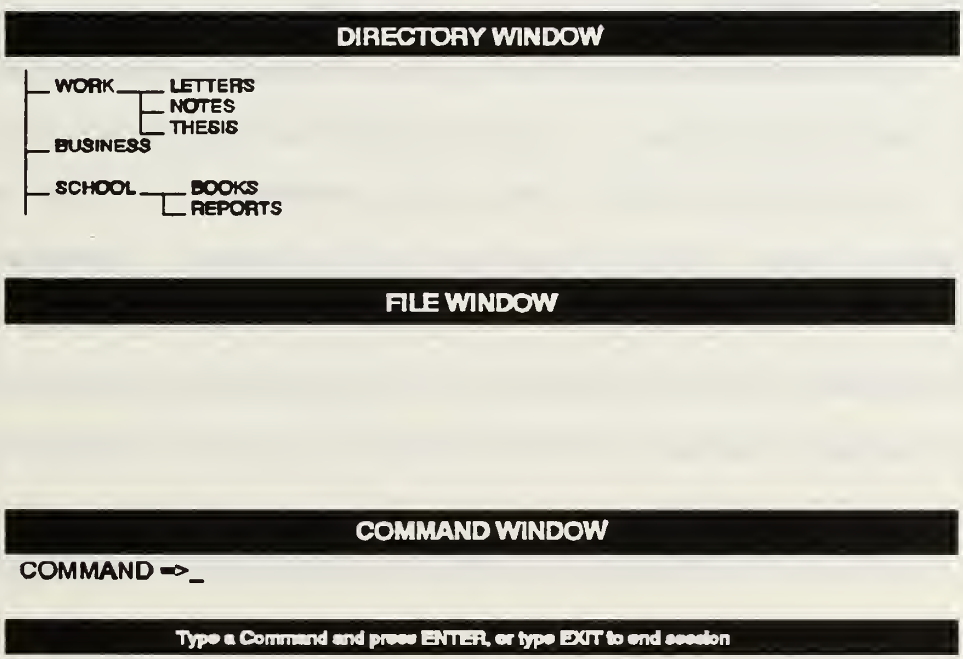

Figure 2. The Command Language Interface Windows.

3. Command Window

The Command Window allows the user to input specific commands to the interface. Each command has a specific syntax associated with it. If the command is typed incorrectly or missing arguments, an error message will appear below the command line indicating the reason for the error.

B. COMMANDS

All commands available for the Command Language Interface are identified below. Commands must be typed in lower case; the <> indicate that the user provides a file name or directory name. The <> are not to be typed. The path for a file or directory is given using the \ and must be included in the command. If a directory is not specified, the interface will assume the root "\" directory is being referenced.

1. Copy File

A file can be copied from one directory to another using the following command:

copyfile <directory\filename> to <directory\filename>

with the user providing the path for the existing file and new file.

2. Create Directory

A new directory can be created using the following command:

createdir <directory\new directory name>

with the user providing the path and name for the new directory.

3. *Create File*

A new file can be created using the following command:

createfile <directory\new file name>

with the user providing the path and name for the new file.

4. *Directory Tree*

A directory tree of a specified directory can be displayed on the screen using the following command:

dirtree <directory>

with the user providing the path of the directory to be displayed.

5. *Help*

The help command provides the user information regarding the interface and any command. At the command line the user can type **help** for general information or the following for more specific information about a command:

help <command>

6. *Listing Directory Files*

The files for a specific directory can be displayed using the following command:

listfiles <directory>

with the user providing the path for the directory to be displayed.

7. *Remove Directory*

When a directory **does not contain files** nor **subdirectories**, and is no longer needed, it can be deleted using the following command:

```
removedir <directory>
```

with the user providing the path of the directory to be deleted.

8. *Remove File*

A file can be deleted from a directory using the following command:

```
removefile <directory\filename>
```

with the user providing the path of the file to be deleted.

9. *Rename a File*

A file can be renamed using the following command:

```
renamefile <directory\filename> to <directory\new
filename>
```

with the user providing the paths for the old and new file names.

10. *Sort Files*

The files in the file window can be sorted by name, date, or size using the following command:

```
sortfile <directory> by date
sortfile <directory> by size
sortfile <directory> by name
```

with the user providing the path for the directory to be sorted.

V. YOUR TASK

Your task is to practice each operation specified below. It is important for you to understand each operation, because the step by step procedures will not be available during the actual experiment. Use the online help and directory tree as necessary. For data collection purposes, it is important that you conduct the experiment without delay.

VI. PRACTICE

A. FILE OPERATIONS:

NOTE: REPEAT EACH OPERATION UNTIL SUCCESSFULLY COMPLETED AND YOU FEEL COMFORTABLE ABOUT THE OPERATION.

1. HELP

To obtain help for the Create File command, enter the following at the command line:

```
help createfile
```

2. CREATE FILE

Create a file named *bobcat* in the *animals* directory:

```
createfile animals\bobcat
```

3. SORT FILES

Sort the *special* directory by size:

```
sortfile special by size
```

4. DELETE FILE

Delete a file named *presnt1* in the *people* directory:

```
removefile people\presnt1
```

5. COPY FILE

Copy the *cat* file in the *animals* directory to *kitty* in the *animals* directory:

```
copyfile animals\cat to animals\kitty
```

6. RENAME FILE

Rename the *girl* file in the *people* directory to *woman* in the *people* directory:

```
renamefile people\girl to people\woman
```

B. DIRECTORY OPERATIONS:

1. CREATE DIRECTORY

Create a subdirectory of *animals* called *mammals*:

createdir animals\mammals

2. TREE

Display a directory tree on the root "\" directory:

dirtree

3. DELETE DIRECTORY

Delete the subdirectory of *animals* called *mammals*:

removedir animals\mammals

****** STOP ******

***** CALL THE EXPERIMENTER TO PROCEED WITH THE EXPERIMENT *****

VII. COMMAND LANGUAGE EXPERIMENT

YOUR NAME: ______________________________

SMC NO: ______________________________

Complete the following operations using the procedures you have learned. Use the help screen as needed. Work at a normal pace and as accurately as possible.

1. Create a subdirectory of \ called *plots*.

2. Create a file called *twoplots.drs* in the *plots* directory.

3. Delete the file called *project.bak* in the *project* directory.

4. Copy the *plot.drs* file in the \ directory to *plot.bak* in the \ directory.

5. Rename the file called *twoplots.drs* in the *plots* directory to *twoplot.bak*.

***** STOP *****

***** CALL THE EXPERIMENTER, TO PROCEED WITH THE EXPERIMENT *****

VIII. COMMAND LANGUAGE EXPERIMENT

Complete the following operations using the procedures you have learned. Use the help screen as needed. Work at a normal pace and as accurately as possible.

1. Copy the *package* file in the *business* directory to the *box* file in the *business* directory.

2. Rename the *papers* file in the *supplies* directory to *document* in the *supplies* directory.

3. Create a file called *car* in the *ground* directory and sort *ground* files by file size.

4. Delete the *planes* directory.

5. Find the largest file of **all** the directories and rename the file to *large.fil*.

IX. COMPLETE THE FOLLOWING QUESTIONS

1. How easy was this interface to use?

(1	2	3	4	5	6	7	8	9)
not at all easy								very easy

2. How easy was this interface to learn?

(1	2	3	4	5	6	7	8	9)
not at all easy								very easy

3. To what extent were the error messages helpful?

(1	2	3	4	5	6	7	8	9)
not at all helpful								very helpful

4. How rapidly did you progress through this experiment?

(1	2	3	4	5	6	7	8	9)
not at all rapid								very rapid

5. How close do you feel you were to 100% accuracy? %________ (consider all errors)

   Estimate the number of errors that occurred ____

6. At any time during the experiment, did the interface appear so difficult that you were ready to abandon the experiment?

   Yes No

   If so, at what point? ______________________________

X. **PLEASE COMPLETE THE FOLLOWING QUESTIONS**

1. Prior to this experiment what was your experience with computers?

   (1 2 3 4 5 6 7 8 9)
   no experience — very experienced

2. If you have an experience level of 2 or greater from the previous question, what type of operating system have you obtained the majority of your experience?

   ____ DOS (Disk Operating System)

   ____ UNIX

   ____ Apple / MacIntosh

   ____ VAX

   ____ Other (Please specify) ____________________

3. Complete the following using the knowledge for the operating system(s) specified above. Please specify the command if applicable.

   a. How do you change directories, i.e., to the "wp" directory?

   ____

   b. How do you list the files in the WP directory?

   ____

   c. How do you erase the "Miscellaneous" file from the WP directory?

   ____

   d. How do you find all batch files in the root directory?

   ____

e. How do you copy all batch files to a floppy disk?

____

f. How do you rename the *autoexec* batch file to a backup file?

____

g. How do you create a new *autoexec* batch file?

____

h. How do you change a file to "read only"?

____

i. How do you convert an *.exe* file to a *.com* file?

____

4. What is your typing experience:

(1 2 3 4 5 6 7 8 9)

no experience — very experienced

Please complete the following:

Age _____

Sex Male Female

Undergraduate degree _______________

NPS Curriculum (if student) _______________

Years of computer experience _______________

APPENDIX C

SUMMARY OF INTERFACE COMMANDS

Command	CLI typed Commands	DMI Steps to Execute Command
Copy File	**copyfile** <directory/filename> **to** <directory/filename>	1. Select **help** icon. 2. Frame Help window.
Create File	**createfile** <directory/filename>	1. Select **directory** icon. 2. Select **new name** of file. 3. Select **create file** icon. 4. Select **accept** from Prompter window.
Create Sub-directory	**createdir** <directory/new subdirectory>	1. Select **directory** to contain new sub-directory. 2. Select **new name** for directory. 3. Select **create directory** icon. 4. Select **accept** from Prompter window.
Delete Directory	**removedir** <directory>	1. Select **directory** to be deleted. 2. Select **delete directory** icon. 3. Select **accept from** Prompter window.

Delete File	**removefile** <directory/filename>	1. Select directory containing file to be deleted. 2. Select file to be removed. 3. Select **delete file** icon. 4. Select **accept** from Prompter window.
Directory Tree	**dirtree**	1. Select **directory tree** icon. 2. Frame Directory Tree window.
Help	**help**	1. Select **help** icon. 2. Frame Help window.
Help on a command	**help** <command>	1. Select **help** icon. 2. Frame Help window. 3. Select command.
List Files in a directory	**listfiles** <directory>	1. Select directory to list files.
Sort File by method (name, date, or size)	**sortfile** <directory> **by** <method>	1. Select directory to be sorted. 2. Select menu for Sort window. 3. Select method for files to be sorted.

APPENDIX D

DIRECT MANIPULATION INTERFACE

YOUR NAME: ______________________________

SMC NO: ______________________________

I. INTRODUCTION

The exercise you are about to participate in involves operating and evaluating a recently-developed computer file management system. You will be asked to read a short description of file management, followed by instructions for each file management operation. You will then practice each operation until you can perform the operation successfully. Upon completing all the operations successfully, you will do a series of tasks that involve using the operations you have learned.

PRIVACY ACT

The information accompanying this experiment will be used for data collection and correlation purposes only. Information provided is voluntary.

II. QUESTIONNAIRE

INSTRUCTIONS:

Please place a T(true) or F(false) next to each of the following statements as you feel it best applies to you.

_____ 1. I enjoy doing work that requires the use of words.

_____ 2. My daydreams are sometimes so vivid I feel as though I actually experience the scene.

_____ 3. I enjoy learning new words.

_____ 4. I can easily think of synonyms for words.

_____ 5. My powers of imagination are higher than average.

_____ 6. I seldom dream.

_____ 7. I read rather slowly.

_____ 8. I cannot generate a mental picture of a friend's face when I close my eyes.

_____ 9. I don't believe that anyone can think in terms of mental pictures.

_____ 10. I prefer to read instructions about how to do something rather than have someone show me.

_____ 11. My dreams are sometimes vivid.

_____ 12. I have better than average fluency in using words.

_____ 13. My daydreams are rather indistinct and hazy.

_____ 14. I spend very little time attempting to increase my vocabulary.

_____ 15. My thinking often consists of mental pictures or images.

III. FILE MANAGEMENT SYSTEM

A. OPERATING SYSTEM

An **operating system** is the software program that makes the hardware useable. The operating system can accomplish many functions such as communicating between the user and the computer (known as the user interface), sharing hardware among users, allowing users to share data among themselves, and many other functions.

An operating system's primary duty is to manage various **files**. Before we go into the details of files, it is important to understand the structure and organization of files.

B. DIRECTORIES AND FILES

A **software program** usually consists of several files. These files work together to produce the program that the user sees and interacts with on the screen. A **directory** contains all of the files used for a given program. For instance, all the files that operate the popular word processing program, WordPerfect 5.1, might be located in a directory called WP51. It is also possible (and highly recommended) to have **subdirectories** within a directory to segregate your files further. For instance, in the WP51 directory you might have a subdirectory titled WORK for all the files relating to your work and one titled THESIS for all the files relating to your thesis.

Figure 1 shows the relationship of this WP51 directory to its subdirectories and files, and also its relationship with the top-level **Root** directory, often represented by a slash (\). Two other directories at the same level as WP51 (here called DOS and HG) also are shown, along with their files.

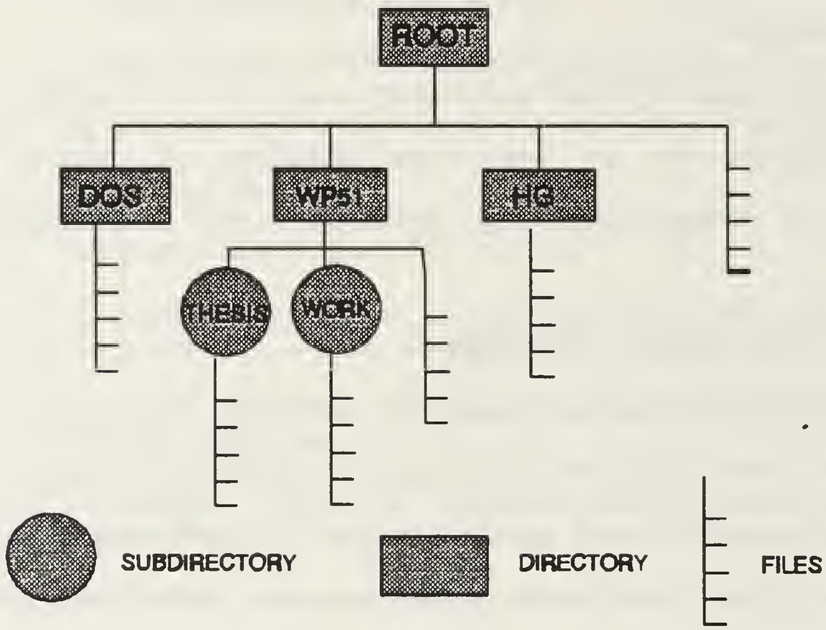

Figure 1. Relationships of the WP51 Directory with other Directories and Files.

It is often necessary to *copy* or *move* a file from one directory to another. The operating system must provide methods to move these files. For example, suppose you wrote a paper for a National Security class titled **<u>The Middle East</u>** and stored this file in the WORK directory under WP51. Later

in the year you decide to use this paper in your thesis and need to transfer the file to the THESIS subdirectory. Some form of *move* or *copy* command would allow you to transfer this file. Additionally, you may want to change the file name to **Chapter II**. Again the operating system interface must provide a means of doing so.

As mentioned earlier, directories contain related files. The operating system interface also must provide a means to **manage** directories. That is, commands such as *create directory* and *delete directory* are needed.

IV. OPERATING SYSTEM INTERFACE

The experimental operating system interface you will be working with is called a **Direct Manipulation Interface**. It uses the mouse device to "click" on to various icons, directory names, or file names on the screen. An **icon** is a graphical representation of an object or an operation. Once activated, the icon carries out a specific function or operation, such as copying a file from one directory to another.

A. MOUSE

The mouse (Figure 2) is the only input device you will be using for this experiment. The keyboard will only be operational when entering your name and SMC at the beginning of the experiment.

Hold the mouse in the right hand (if right handed, left hand if left handed) so that the cord and buttons are at the top. Place your index and middle fingers lightly over the mouse buttons. Gently guide the movement of the mouse with the hand.

Normally, the mouse is represented as an arrow, or cursor, which moves on the screen as you move the mouse. When the system is performing an operation that takes longer than one second, the cursor will appear as an hour glass to indicate that the operation will take time. The cursor is not functional in the hour glass mode.

Gently press or "click" the **left** button to **select** the item on the screen superimposed by the arrow or cursor. Click the **right** button to call up a menu that describes operations you can do in a given window. Each window has a menu assigned to it. Most of the window menus are not operational. However, the **help**, **sort**, **tree**, and **error** windows do have operational menus.

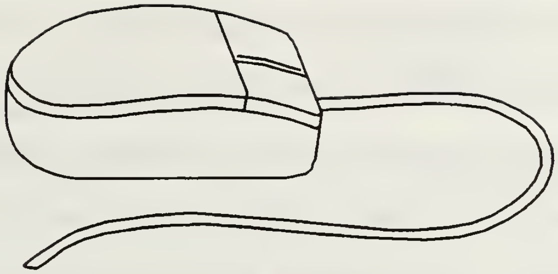

Figure 2. Mouse Control Device

B. WINDOWS

The Direct Manipulation Interface consists of five windows (Figure 3). These are (1) Directory Window, (2) File Window, (3) File Sort Window, (4) New Name Window, and (5) Icon Window. Each window has a specific function and interacts with the other windows by use of mouse operations. Additionally, the Help and Tree windows will pop up onto the screen when the icon that represents one of them is selected.

All windows have similar characteristics. The **Top Pane** is the main window that includes all other windows and encompasses the entire screen. The **label** of the top pane contains the name of the interface or the name of a selected directory. The **Directory**, **File**, and **New Name windows** all operate in a similar manner when the user selects an item. The selected item will appear highlighted. Due to the limited

size of the windows, not all files, directories, or new names may fit in the window at one time. The window operations provide the ability to **scroll** the window. Scroll a window by pressing and holding the right mouse button (cursor changes to a four directional arrow) in the window to be scrolled. Move the cursor out of the window in the direction of the additional names. The **scroll bar** to the right of the window shows the status of the scrolling with respect to the total list. The scroll bar appears only during the scrolling operation.

1. *Directory Window*

The Directory window contains a list of all the directories on the disk in alphabetical order. The directories are indented to reflect their hierarchical or tree structure. For example, each subdirectory will be indented one space from the parent directory.

2. *File Window*

The name of the selected directory appears in reverse video (black background, white lettering) in the Directory window and the names of all files contained in the directory are displayed in the File window. Files listed in the File window are in alphabetical order. All operations on files will be carried out using the File window.

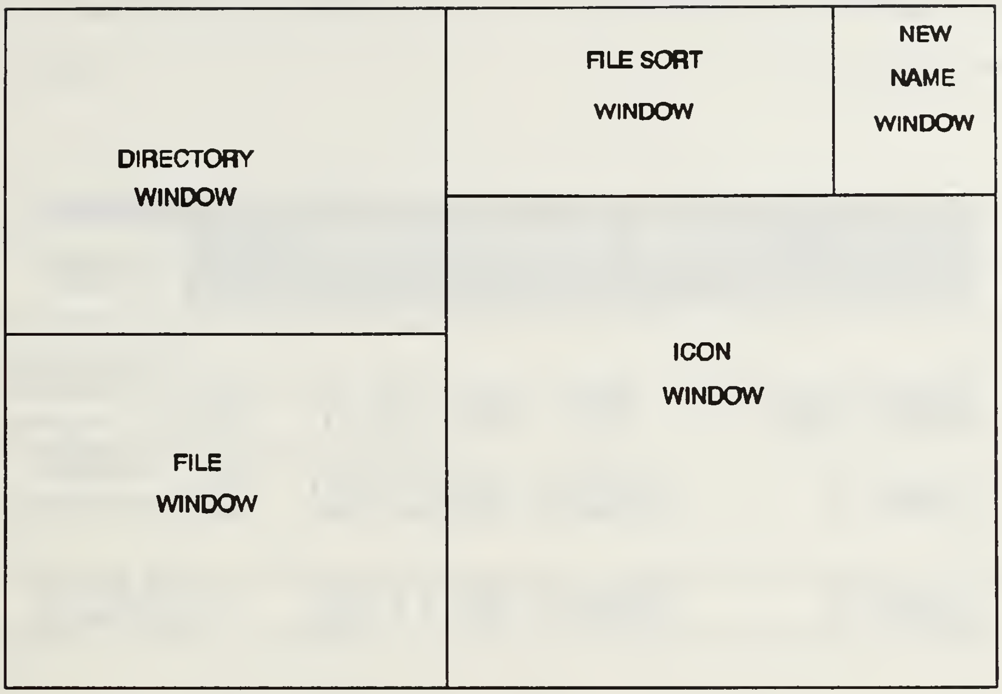

Figure 3. The Direct Manipulation Interface Top Pane and Component Windows

3. *File Sort Window*

The File Sort window contains the names of all the files listed in the File window, plus additional information about these files. The File Sort window has two parts, a label and a text portion. The label, i.e., Files Sorted by Name, contains a menu which allows the files to be sorted by name, date, or size. Items can be selected from this menu by clicking the right mouse button when the arrow is over the label and selecting with the left mouse button the desired

sorting method. The File Sort window will then contain the files of the selected directory sorted by the specified method (see Figure 4).

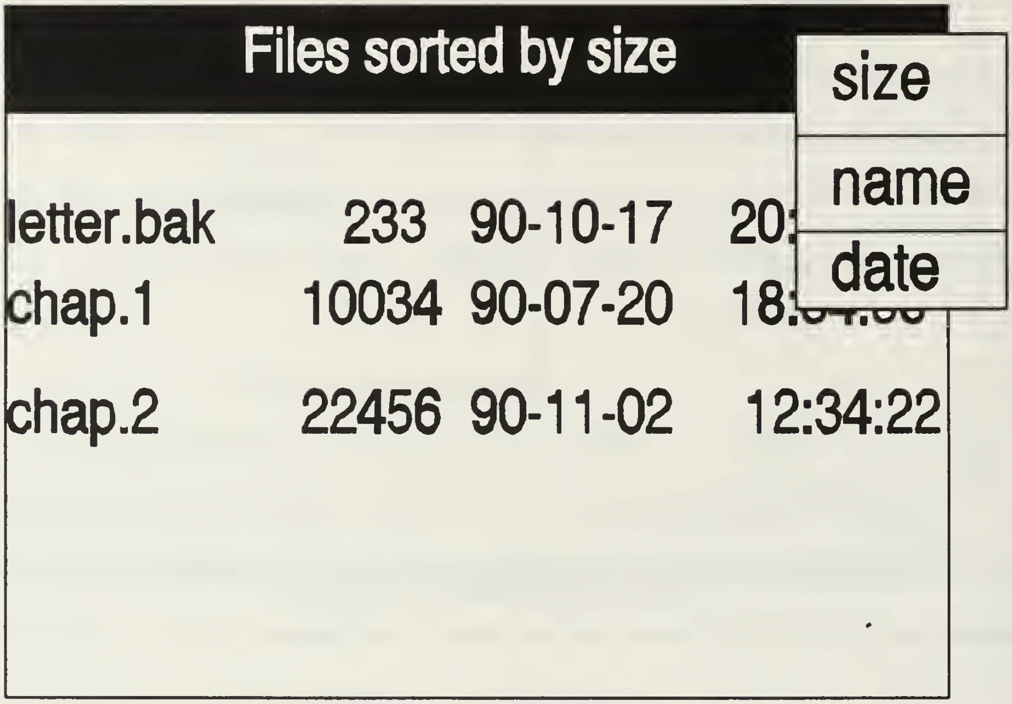

Figure 4. Example of File Sort Window, with File Names Sorted by Size.

4. New Name Window

The New Name window contains a list of all the names needed for new files, renamed files, and directories. The new name must be selected before an icon is selected to complete

an operation on a file, if that operation involves naming the file. The system removes the selected new name after it has been used.

5. *Icon Window*

The Icon window is divided into three sections: File Icons, Directory Icons, and Other Icons. Using this window, files and directories can be created, copied, renamed, and deleted, as discussed below.

a. *Create File Icon*

The **Create File** icon creates a new file in the selected directory using the name selected in the New Name window.

b. *Copy File Icon*

The **Copy File** icon copies a selected file to the selected directory using the selected name in the New Name window.

c. *Rename File Icon*

The **Rename File** icon renames a selected file to the selected name in the New Name window.

d. *Delete File Icon*

The **Delete File** icon deletes the selected file from the selected directory.

e. *Create Directory Icon*

The **Create Directory** icon creates a new subdirectory under the selected directory using the selected name in the New Name window.

f. Directory Tree

The **Directory Tree** icon displays a graphical depiction of the directories on the drive.

g. Delete Directory

The **Delete Directory** icon deletes the selected directory from the disk. The selected directory **cannot** contain **files** nor **subdirectories**.

h. Help

The **Help** icon displays the Help window described below.

6. Prompter Window

When you are performing a file or directory operation a "Prompter window" will appear. This window allows you to confirm or cancel the operation. When this window appears, click the **right** mouse button while the arrow is in the white portion of the window (suggested file or directory name) to call up the menu. Then select the "accept" or "cancel" option with the **left** mouse button. A Prompter window also will appear when you attempt to conduct an operation without having specified all the necessary information. The needed information will appear in the white portion of the window. Remove the Prompter window by selecting "cancel" from the prompter menu.

7. Help Window

The Help window provides information on window operations, icons, and window locations. The Help window can be displayed by selecting the *Help* icon. **Hold down** the left

mouse button until a prompt appears for the left corner of the Help window. **Move** the pop-up window corner to the upper left corner of the screen and release the mouse button. The lower right corner of the window will appear and can be repositioned with the mouse. Click the **left** mouse button when the size of the window is at least 5 inches square.

The help commands appear in alphabetical order. When a command is sellected, a description of the command will be provided in the right pane (text pane) of the window. The command list and text pane can be scrolled in the manner discussed earlier: press the **right** mouse button and **drag** the cursor out of the pane in the direction of the unseen text.

8. Error Windows

Error windows can appear either as a Prompter window or as a Pop-up window. A Prompter window will be a small window with a short message. Selecting an icon without all the necessary information specified will cause it to appear. It can be removed by (1) selecting the **right** mouse button when the cursor is located in the white portion of the prompter (area of short message) to obtain the menu options and then (2) selecting the "cancel" option.

If you attempt an operation that the operating system does not allow, an Error Pop-up window will appear in the middle of the screen. The label, located at the top, will contain the error message. The remaining text portion of the window may be confusing and it is not necessary for you to

understand it. To remove the window, use the mouse's **left** button and cursor to select a point outside the window, **or** select the menu with the **right** mouse button in the label and then the "close" option with the **left** mouse button.

9. Directory Tree

A Directory Tree is helpful for seeing the "whole picture" of directories and subdirectories. As in the Directory window, a subdirectory will branch off from its parent directory. The Directory Tree window can be displayed by selecting the *Tree* icon. **Hold down** the **left** mouse button until a prompt appears at the left corner of the Tree window. **Move** the Pop-up window corner to the upper left corner of the screen and release the mouse button. The lower right corner of the window will appear and can be repositioned with the mouse. Click the **left** mouse button when the size of the window is approximately 5 to 6 inches square.

V. YOUR TASK

Your task is to practice each operation specified below. It is important for you to understand each operation, because the step by step procedures will not be available during the actual experiment. Use the online help and Directory Tree as necessary. For data collection purposes, it is important that you conduct the experiment without delay.

VI. PRACTICE

A. FILE OPERATIONS:

NOTE: REPEAT EACH OPERATION UNTIL IT IS SUCCESSFULLY COMPLETED AND YOU FEEL COMFORTABLE ABOUT THE OPERATION.

1. HELP SCREEN
   a. Select the **Help** icon, holding the left mouse button down.
   b. Move the cursor to the upper left corner of the screen and release the mouse button.
   c. Move the cursor to the lower right portion of the screen and press the left mouse button.
   d. Select the Create **File** item for information.
   e. To close Help window, press the right mouse button on the window label.
   f. Select close.

2. CREATE FILE
   a. Select the *animals* directory.
   b. Select the *bobcat* name from the New Name window.
   c. Select the **Create File** icon.
   d. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   e. Select "accept" from the Prompter window menu.

3. SORT FILES
   a. Select the *special* directory from the Directory window.
   b. Select the menu for the Sort window by pressing the right mouse button over the label for the Sort window.
   c. Select **Size** order in the Pop-up menu.

4. DELETE FILE
   a. Select the *people* directory from the Directory window.
   b. Select the *presnt1* file from the File window.
   c. Select the **Delete File** icon.
   d. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   e. Select "accept" from the Prompter window menu.

5. COPY FILE
   a. Select the *animals* directory from the Directory window.
   b. Select the *cat* file from the File window.
   c. Select the *kitty* name from the New Name window.
   d. Select the **Copy** icon.
   e. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   f. Select "accept" from the Prompter window menu.

6. RENAME FILE
   a. Select the *people* directory from the Directory window.
   b. Select the *girl* file from the File window.
   c. Select the *woman* name from the New Name window.
   d. Select the **Rename File** icon.
   e. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   f. Select "accept" from the Prompter window menu.

B. DIRECTORY OPERATIONS:

1. CREATE DIRECTORY
   a. Select the *animals* directory from the Directory window.
   b. Select the New Name *mammals* from the New Name window.
   c. Select the **Create Directory** icon.
   d. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   e. Select "accept" from the Prompter window menu.

2. TREE
   a. Select the **Tree** icon, hold the left mouse button down and move the cursor to the upper left corner of the screen and release.
   b. When the lower right corner of the window appears, position it to the lower right portion of the screen and click the left mouse button.
   c. To close the Directory Tree, press the right mouse button on the window label.
   d. Select close.

3. DELETE DIRECTORY
   b. Select the *mammals* directory from the Directory window (NOTE: there must be no files present in the directory to be deleted).
   c. Select the **Delete Directory** icon.
   d. Select the Prompter menu by pressing right mouse button in the white portion of the Prompter window.
   e. Select "accept" from the Prompter window menu.

****** STOP ******

***** CALL THE EXPERIMENTER, TO PROCEED WITH THE EXPERIMENT *****

VII. DIRECT MANIPULATION EXPERIMENT

YOUR NAME: ______________________________

SMC NO: ______________________________

Complete the following operations using the procedures you have learned. Use the help screen as needed. Work at a normal pace and as accurately as possible.

1. Create a subdirectory of \ called *plots*.

2. Create a file called *twoplots.drs* in the *plots* directory.

3. Delete the file called *project.bak* in the *project* directory.

4. Copy the *plot.drs* file in the \ directory to *plot.bak* in the \ directory.

5. Rename the file called *twoplots.drs* in the *plots* directory to *twoplot.bak*.

***** STOP *****

***** CALL THE EXPERIMENTER, TO PROCEED WITH THE EXPERIMENT *****

VIII. DIRECT MANIPULATION EXPERIMENT

Complete the following operations using the procedures you have learned. Use the help screen as needed. Work at a normal pace and as accurately as possible.

1. Copy the *package* file in the *business* directory to the *box* file in the *business* directory.

2. Rename the *papers* file in the *supplies* directory to *document* in the *supplies* directory.

3. Create a file called *car* in the *ground* directory and sort *ground* files by file size.

4. Delete the *planes* directory.

5. Find the largest file of **all** the directories and rename the file to *large.fil*.

IX. COMPLETE THE FOLLOWING QUESTIONS

1. How easy was this interface to use?

(1	2	3	4	5	6	7	8	9)
not at all easy								very easy

2. How easy was this interface to learn?

(1	2	3	4	5	6	7	8	9)
not at all easy								very easy

3. To what extent were the error messages helpful?

(1	2	3	4	5	6	7	8	9)
not at all helpful								very helpful

4. How rapidly did you progress through this experiment?

(1	2	3	4	5	6	7	8	9)
not at all rapid								very rapid

5. How close do you feel you were to 100% accuracy? %_______ (consider all errors)

   Estimate the number of errors that occurred ____

6. At any time during the experiment, did the interface appear so difficult that you were ready to abandon the experiment?

   Yes   No

   If so, at what point? ______________________________

X. PLEASE COMPLETE THE FOLLOWING QUESTIONS

1. Prior to this experiment what was your experience with computers?

   (1 2 3 4 5 6 7 8 9)
   no experience ... very experienced

2. If you have an experience level of 2 or greater from the previous question, what type of operating system have you obtained the majority of your experience?

   ____ DOS (Disk Operating System)

   ____ UNIX

   ____ Apple / MacIntosh

   ____ VAX

   ____ Other (Please specify) ____________________

3. Complete the following using the knowledge for the operating system(s) specified above. Please specify the command if applicable.

   a. How do you change directories, i.e., to the "wp" directory?

   ____

   b. How do you list the files in the WP directory?

   ____

   c. How do you erase the "Miscellaneous" file from the WP directory?

   ____

   d. How do you find all batch files in the root directory?

   ____

e. How do you copy all batch files to a floppy disk?

____

f. How do you rename the *autoexec* batch file to a backup file?

____

g. How do you create a new *autoexec* batch file?

____

h. How do you change a file to "read only"?

____

i. How do you convert an *.exe* file to a *.com* file?

____

4. What is your experience with using a mouse:

(1	2	3	4	5	6	7	8	9)
no experience								very experienced

Please complete the following:

Age ______

Sex Male Female

Undergraduate degree ______________

NPS Curriculum (if student) ______________

Years of computer experience ______________

APPENDIX E

EXAMPLE OF CLI USER LOG

```
Smith, John
1000
1-24-1991
CLI
simple
15:31:19 DIRC createdir plots
15:32:15 FLCR createfile twoplots.drs
15:33:10 ERRS deletefile project.bak
15:33:45 ERRS delfile project.bak
15:33:56 HELP help
15:34:58 RMF3 removefile project.bak
15:36:13 HELP help
15:36:28 HELP help
15:37:17 HELP help removefile
15:38:14 FILR removefile project\project.bak
15:39:20 FILC copyfile plot.drs to plot.bak
15:39:32 ERRS elp
15:39:41 HELP help
15:41:43 RNF1 renamefile poots\twoplots.drs
         toplots\twoplot.bak
15:42:07 HELP help renamefile
15:43:26 RNF4 renamefile plots\twoplots.drs to
         plots\twoplot.bak
15:44:22 HELP help renamefile
15:45:50 RNF4 renamefile plots\twoplots.drs to
         plots\twoplot.bak
15:46:23 DRTS dirtree
15:47:57 RNF4 renamefile plots\twoplots.drs to
         plots\twoplot.bak
15:48:36 ERRS listfile
15:48:49 HELP help
15:49:17 LSFD listfiles plots
15:49:59 LSFD listfiles plots
```

APPENDIX F

EXAMPLE OF DMI USER LOG

```
Doe, John
2000
1-25-1991
DMI
simple
10:52:48   SDN    b:
10:54:18   SDN    b:\
10:54:25   SNN    plots
10:54:40   SDN    b:
10:54:40   SDN    b:\plots
10:54:40   CDS    Create directory  b:\plots
10:55:11   SNN    twoplots.drs
10:55:20   CFS    Create New File
                  b:\plots\twoplots.drs
10:55:22   SDN    b:\plots
10:55:29   SDN    b:\project
10:55:36   SFN    project.bak
10:55:45   DFS    Delete file  \project\project.bak
10:55:47   SDN    b:\project
10:56:16   HLP    Help window displayed
10:56:24   HLP    Help window displayed
10:57:03   SDN    b:\
10:57:34   SFN    plot.drs
10:58:19   PF2    ERROR copy file without selectedName
10:58:30   PF2    ERROR copy file without selectedName
10:58:37   SNN    plot.bak
10:59:03   SNN    plot.bak
10:59:05   SNN    plot.bak
10:59:20   WBW    WALKBACK window generated: message
                  not understood: asLowerCase
10:59:58   SDN    b:\
10:59:59   PFS    Copy file b:\plot.drs to b:\plot.bak
11:00:30   PF2    ERROR copy file without selectedName
11:00:31   HLP    Help window displayed
11:00:35   HLP    Help window displayed
11:00:43   SHP    Copy File
11:01:25   SDN    b:\plots
11:01:39   SFN    twoplots.drs
11:01:43   SNN    twoplot.bak
11:01:52   RFS    Rename  file  b:\plots\twoplots.drs
                  to b:\plots\twoplot.bak
```

APPENDIX G

CODES AND MESSAGES FOR CLI

CODE	MESSAGE
CRD1	Error in syntax for CREATEDIR. Type HELP CREATEDIR for correct syntax
CRD2	Directory name is too long: please specify another name, 8 characters or less
CRD3	This directory already exists: please specify another name
CRD4	A file by that name already exists: please provide another name
DIRC	Directory created
RMD1	Error in syntax for REMOVEDIR. Type HELP REMOVEDIR for correct syntax
RMD2	Directory name is too long: please specify another name, 8 characters or less
RMD3	The directory specified does not exist: please provide another name
RMD4	The name specified exists as a file: you need to provide a directory name
DIRR	Directory removed
CPF1	Error in syntax for COPYFILE. Type HELP COPYFILE for correct syntax
CPF2	File name is too long: please specify another name, 8 characters or less
CPF3	The target file already exists: please specify another name
CPF4	The source file does not exist: please specify another name
FILC	File copied
CRF1	Error in syntax for CREATEFILE. Type HELP CREATEFILE for correct syntax
CRF2	File name is too long: please specify another name
CRF3	The file already exists: please specify another name
CRF4	The directory name is not valid: please specify another name
FLCR	File created
RNF1	Error in syntax for RENAMEFILE. Type HELP RENAMEFILE for correct syntax
RNF2	File name is too long: please specify another name, 8 characters or less
RNF3	The target file already exists: please specify another name
RNF4	The source file does not exist: please specify another name
FLRN	File renamed

APPENDIX G (continued)

RMF1	Error in syntax for REMOVEFILE. Type HELP REMOVEFILE for correct syntax
RMF2	File name is too long: please specify another name, 8 characters or less
RMF3	The file does not exist: please specify another name

RMF4	The directory name is not valid: please specify another name
FILR	File removed
LSF1	Error in syntax for LISTFILES. Type HELP LISTFILES for correct syntax
LSF2	Directory name is too long: please specify another name, 8 characters or less
LSF3	The directory specified does not exist: please provide another name
LSFD	List of files in directory displayed
SRF1	Error in syntax for SORTFILE. Type HELP SORTFILE for correct syntax
SRF2	Directory name is too long: please specify another name, 8 characters or less
SRF3	The directory specified does not exist: please provide another name
SRFD	Sorted list of files in directory displayed
DRT1	Error in syntax for DIRTREE. Type HELP DIRTREE for correct syntax
DRTS	Directory tree displayed

APPENDIX H

CODES AND MESSAGES FOR DMI

CODE	MESSAGE
CD1	ERROR create directory without selected directory
CD2	ERROR create directory without selected name
CDS	Create directory successful
DD1	ERROR delete directory without selected directory
DDS	Delete directory successful
PF1	ERROR copy file without selected file
PF2	ERROR copy file without selected name
PFS	Copy file successful
CF1	ERROR create file without selected directory
CF2	ERROR create file without selected name
CFS	Create file successful
RF1	ERROR rename file without selected file
RF2	ERROR rename file without selected name
RFS	Rename file successful
DF1	ERROR delete file without selected file
DFS	Delete file successful
WBW	ERROR SmallTalk WalkBack Window generated due to DOS error
SDN	Select directory name
SNN	Select new name
SFN	Select file name
SHP	Select help
HLP	Select help object
SPS	Sort files successful
DTS	Directory tree displayed

REFERENCES

1. Card, S., Moran, T., and Newell, A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.

2. Rasmussen, J., and Zunde, P., *Empirical Foundations of Information and Software Science III*, Plenum Press, 1987.

3. Kieras, D., and Polson, P., "An Approach to the Formal Analysis of User Complexity," *International Journal of Man-Machine Studies*, vol. 22, no. 4, April 1985.

4. Bovair, S., Kieras, D., and Polson, P., "The Acquisition and Performance of Text-Editing Skill: A Cognitive Complexity Analysis," *Human Computer Interaction*, vol. 5, no. 3, 1990.

5. Campbell, D., "Task Complexity: A Review and Analysis," *Academy of Managment Review*, vol. 13, no. 1, 1988.

6. Wood, R., "Task Complexity: Definition of the Construct," *Organizational Behavior and Human Decision Processes*, vol. 37, 1987.

7. Margono, S., and Shneiderman, B., "A Study of File Manipulation by Novices Using Commands versus Direct Manipulation," *26th Annual Technical Symposium*, June 1987.

8. Jacob, R., J. K., "A Specification Language for Direct-Manipulation User Interfaces," *ACM Transactions on Graphics*, vol. 5, no. 4, Oct 1986.

9. Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Company, 1987.

10. Karat, J., "Evaluating User Interface Complexity," *Human Computer Interaction - INTERACT '87*, 1987.

11. Te'eni, D., "Direct Manipulation as a Source of Cognitive Feedback: A Human-computer Experiment with a Judgement Task," July 1989.

12. Rogers, Y., "Icons at the Interface: Their Usefulness," *Interacting with Computers*, vol. 1, no. 1, 1989.

13. Richardson, A., "Verbalizer-Visualizer: A Cognitive Style Dimension," *Journal of Mental Imagery*, vol. 1, no. 1, 1977.

14. Winer, B. J., *Statistical Principles and Experimental Design*, McGraw-Hill Book Co., 1971.

15. *SAS/STAT Guide for Personal Computers*, Version 6, N. C. Cary, ed., SAS, 1987.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	LT Nancy A. Reinhard CNO OP64D2 Pentagon Room 4D581 Washington, DC 20350-2000	3
4.	Prof. Kishore Sengupta Code AS/SE Naval Postgraduate School Monterey, California 93943	3
5.	Prof. Judith Lind Code OR/LI Naval Postgraduate School Monterey, California 93943	2
6.	Prof. Tung Bui Code AS/BD Naval Postgraduate School Monterey, California 93943	1
7.	CDR Richard Miller NDU IMDIRECTORATE Fort McNair Washington, DC 20319-6000	1
8.	Dr. Marion Kibbe Code 3152 Naval Weapons Center China Lake, California 93555	1

Thesis
R32983 Reinhard
c.1 The effect of task complexity on user interfaces.